Assignment_1-Yiran

January 22, 2020

1 Assignment 1 - Probability, Linear Algebra, Programming, and Git

1.1 Yiran Chen

Netid: yc390

Instructions for all assignments can be found here, which is also linked to from the course syllabus.

2 Probability and Statistics Theory

Note: for all assignments, write out all equations and math using markdown and LaTeX. For this section of the assignment (Probability and Statistics Theory) show and type up ALL math work

2.1 1

[3 points]

Let $f(x) = \begin{cases} 0 & x < 0 \\ \alpha x^2 & 0 \leq x \leq 2 \\ 0 & 2 < x \end{cases}$

For what value of α is $f(x)$ a valid probability density function?

ANSWER

A valid probability density function needs to satisfy:

1. $f(x) >= 0$
2. $\int_{-\infty}^{\infty} f(x)dx = 1$

Therefore,

1. $\alpha x^2 >= 0, \alpha >= 0$
2. $\int_0^2 f(x)dx = \int_0^2 \alpha x^2 dx = \frac{8\alpha}{3} = 1$

$\alpha = \frac{3}{8}$

2.2 2

[3 points] What is the cumulative distribution function (CDF) that corresponds to the following probability distribution function? Please state the value of the CDF for all possible values of x.

$$f(x) = \begin{cases} \frac{1}{3} & 0 < x < 3 \\ 0 & \text{otherwise} \end{cases}$$

ANSWER

For 0 < x < 3, $F(x) = \int_0^3 f(x)dx = \int_0^3 \frac{1}{3}dx = \frac{x}{3}$

$$F(x) = \begin{cases} 0 & x < 0 \\ \frac{x}{3} & 0 \leq x \leq 3 \\ 1 & x > 3 \end{cases}$$

2.3 3

[**6 points**] For the probability distribution function for the random variable X,

$$f(x) = \begin{cases} \frac{1}{3} & 0 < x < 3 \\ 0 & \text{otherwise} \end{cases}$$

what is the (a) expected value and (b) variance of X. *Show all work.*

ANSWER

$\mu = E(x) = \int_{-\infty}^{\infty} xf(x)dx$
$= \int_{-\infty}^{0} xf(x)dx + \int_0^3 xf(x)dx + \int_3^{\infty} xf(x)dx$
$= \int_0^3 \frac{x}{3}dx$
$= \left[\frac{x^2}{6}\right]_0^3$
$= 1.5$

$V(x) = \int_{-\infty}^{\infty}(x-\mu)^2 f(x)dx$
$= \int_{-\infty}^{0}(x-\mu)^2 f(x)dx + \int_0^3 (x-\mu)^2 f(x)dx + \int_3^{\infty}(x-\mu)^2 f(x)dx$
$= \frac{1}{3}\int_0^3 (x-\mu)^2 dx$
$= \frac{1}{3}\int_0^3 (x^2 - 2\mu x + \mu^2)^2 dx$
$= \frac{1}{3}\left[\frac{1}{3}(x-\mu)^3 - \mu x^2 + \mu^2 x\right]_0^3$
$= \frac{1}{3}\left[\frac{1}{3}(x-1.5)^3 - 1.5x^2 + 1.5^2 x\right]_0^3$
$= 0.75$

2.4 4

[**6 points**] Consider the following table of data that provides the values of a discrete data vector $\mathbf{x}$ of samples from the random variable X, where each entry in $\mathbf{x}$ is given as x_i.

Table 1. Dataset N=5 observations

	x_0	x_1	x_2	x_3	x_4
x	2	3	10	-1	-1

What is the (a) mean, (b) variance, and the of the data?

Show all work. Your answer should include the definition of mean, median, and variance in the *context of discrete data.*

ANSWER

(a) mean:

$\mu = \frac{(x_0+x_1+x_2+x_3+x_4)}{N}$
$= \frac{(2+3+10-1-1)}{5}$
$= \frac{13}{5}$

(b) median:

The middle value of the five observations in sorted order is 2.

$median = 2$

(c) variance:

Assuming each observation is collected with equal probability of $p(x_i) = \frac{1}{5}$

$variance = \sum(x_i - \mu)^2 p(x_i)$
$= \frac{1}{5}((2 - \frac{13}{5})^2 + (3 - \frac{13}{5})^2 + (10 - \frac{13}{5})^2 + 2(-1 - \frac{13}{5})^2)$
$= \frac{406}{25}$

2.5 5

[**8 points**] Review of counting from probability theory.

(a) How many different 7-place license plates are possible if the first 3 places only contain letters and the last 4 only contain numbers?

(b) How many different batting orders are possible for a baseball team with 9 players?

(c) How many batting orders of 5 players are possible for a team with 9 players total?

(d) Let's assume this class has 26 students and we want to form project teams. How many unique teams of 3 are possible?

Hint: For each problem, determine if order matters, and if it should be calculated with or without replacement.

ANSWER

(a) order does not matter, with replacement

$26^3 \times 10^4 = 175760000$

(b) order matters, without replacement

$9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362880$

(c) order matters, without replacement

$9 \times 8 \times 7 \times 6 \times 5 = 15120$

(d) order doesn't matter, without replacement

$\binom{26}{3} = \frac{26!}{(26-3)!3!} = 2600$

3 Linear Algebra

3.1 6

[**7 points**] **Matrix manipulations and multiplication**. Machine learning involves working with many matrices, so this exercise will provide you with the opportunity to practice those skills.

Let $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} -1 \\ 3 \\ 8 \end{bmatrix}$, $\mathbf{c} = \begin{bmatrix} 4 \\ -3 \\ 6 \end{bmatrix}$, and $\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Compute the following or indicate that it cannot be computed:

1. $\mathbf{AA}$
2. $\mathbf{AA}^T$
3. $\mathbf{Ab}$
4. $\mathbf{Ab}^T$
5. $\mathbf{bA}$
6. $\mathbf{b}^T\mathbf{A}$
7. $\mathbf{bb}$
8. $\mathbf{b}^T\mathbf{b}$
9. $\mathbf{bb}^T$
10. $\mathbf{b} + \mathbf{c}^T$
11. $\mathbf{b}^T\mathbf{b}^T$
12. $\mathbf{A}^{-1}\mathbf{b}$
13. $\mathbf{A} \circ \mathbf{A}$
14. $\mathbf{b} \circ \mathbf{c}$

Note: The element-wise (or Hadamard) product is the product of each element in one matrix with the corresponding element in another matrix, and is represented by the symbol "∘".

ANSWER

[1]:
```
import numpy as np
A = np.matrix([(1,2,3),
               (2,4,5),
               (3,5,6)])
b = np.array([-1,3,8])
c = np.array([4,-3,6])
I = np.matrix([(1,0,0),
               (0,1,0),
               (0,0,1)])
```

1.

$$\mathbf{AA} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$
$$= \begin{bmatrix} 1\times1+2\times2+3\times3 & 1\times2+2\times4+3\times5 & 1\times3+2\times5+3\times6 \\ 2\times1+4\times2+5\times3 & 2\times2+4\times4+5\times5 & 2\times3+4\times5+5\times6 \\ 3\times1+5\times2+6\times3 & 3\times2+5\times4+6\times5 & 3\times3+5\times5+6\times6 \end{bmatrix}$$

$$= \begin{bmatrix} 14 & 25 & 31 \\ 25 & 45 & 56 \\ 31 & 56 & 70 \end{bmatrix}$$

[2]:
```
print('AA = \n',np.matmul(A,A))
```

```
AA =
 [[14 25 31]
 [25 45 56]
 [31 56 70]]
```

2.

$$\mathbf{A}\mathbf{A}^T = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$
$$= \begin{bmatrix} 1\times1+2\times2+3\times3 & 1\times2+2\times4+3\times5 & 1\times3+2\times5+3\times6 \\ 2\times1+4\times2+5\times3 & 2\times2+4\times4+5\times5 & 2\times3+4\times5+5\times6 \\ 3\times1+5\times2+6\times3 & 3\times2+5\times4+6\times5 & 3\times3+5\times5+6\times6 \end{bmatrix}$$
$$= \begin{bmatrix} 14 & 25 & 31 \\ 25 & 45 & 56 \\ 31 & 56 & 70 \end{bmatrix}$$

[3]:
```
print('A(A^T) = \n',np.matmul(A,A.T))
```

```
A(A^T) =
 [[14 25 31]
 [25 45 56]
 [31 56 70]]
```

3.

$$\mathbf{A}\mathbf{b} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} -1 \\ 3 \\ 8 \end{bmatrix}$$
$$= \begin{bmatrix} 1\times(-1)+2\times3+3\times8 \\ 2\times(-1)+4\times3+5\times8 \\ 3\times(-1)+5\times3+6\times8 \end{bmatrix}$$
$$= \begin{bmatrix} 29 \\ 50 \\ 60 \end{bmatrix}$$

[4]:
```
print('Ab = ', np.matmul(A,b))
```

```
Ab =  [[29 50 60]]
```

4.

$\mathbf{A}\mathbf{b}^T$

$((3\times3)\times(1\times3))$ matrix cannot be computed.

It cannot be computed as $\mathbf{A}$ is 3x3 matrice and $\mathbf{b}^T$ is 1x3 matrice.

5.

bA

$((3 \times 1) \times (3 \times 3))$ matrix cannot be computed.

It cannot be computed as $\mathbf{b}$ is 3x1 matrix and $\mathbf{A}^T$ is 3x3 matrice.

6.

$$\mathbf{b}^T\mathbf{A} = \begin{bmatrix} -1 & 3 & 8 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$
$$= \begin{bmatrix} -1 \times 1 + 3 \times 2 + 8 \times 3 & -1 \times 2 + 3 \times 4 + 8 \times 5 & -1 \times 3 + 3 \times 5 + 8 \times 6 \end{bmatrix}$$
$$= \begin{bmatrix} 29 & 50 & 60 \end{bmatrix}$$

[5]:
```python
print('(b^T)A = ', np.matmul(b.T,A))
```

```
(b^T)A =  [[29 50 60]]
```

7.

bb

$((3 \times 1) \times (3 \times 1))$ matrix cannot be computed.

It cannot be computed as $\mathbf{b}$ is 3x1 matrix.

8.

$$\mathbf{b}^T\mathbf{b} = \begin{bmatrix} -1 & 3 & 8 \end{bmatrix} \times \begin{bmatrix} -1 \\ 3 \\ 8 \end{bmatrix}$$
$$= \begin{bmatrix} -1 \times (-1) + 3 \times 3 + 8 \times 8 \end{bmatrix}$$
$$= \begin{bmatrix} 1 + 9 + 64 \end{bmatrix} = \begin{bmatrix} 74 \end{bmatrix}$$

[6]:
```python
print('(b^T)b = ', np.matmul(b.T,b))
```

```
(b^T)b =  74
```

9.

$$\mathbf{b}\mathbf{b}^T = \begin{bmatrix} -1 \\ 3 \\ 8 \end{bmatrix} \times \begin{bmatrix} -1 & 3 & 8 \end{bmatrix} = \begin{bmatrix} -1 \times (-1) & -1 \times 3 & -1 \times 8 \\ 3 \times (-1) & 3 \times 3 & 3 \times 8 \\ 8 \times (-1) & 8 \times 3 & 8 \times 8 \end{bmatrix} = \begin{bmatrix} 1 & -3 & -8 \\ -3 & 9 & 24 \\ -8 & 24 & 64 \end{bmatrix}$$

[7]:
```python
b = np.matrix([(-1),(3),(8)])
print('b(b^T) = \n', np.matmul(b.T,b))
# normal numpy operation cannot get the correct result
# need to transform b from array to matrix and switch position of b.T and b
```

```
b(b^T) =
 [[ 1 -3 -8]
```

```
[-3  9 24]
[-8 24 64]]
```

[8]:
```
# change b back to numpy array
b = np.array([-1,3,8])
```

10.

$\mathbf{b} + \mathbf{c}^T$

It cannot be computed as $\mathbf{b}$ is 3x1 matrice and $\mathbf{c}^T$ is 1x3 matrice which cannot be added together.

11.

$\mathbf{b}^T\mathbf{b}^T$

$((1 \times 3) \times (1 \times 3))$ matrix cannot be computed.

It cannot be computed as $\mathbf{b}^T$ is 1x3 matrix.

12.

$\mathbf{A}^{-1}$:

$$\begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 2 & 4 & 5 & | & 0 & 1 & 0 \\ 3 & 5 & 6 & | & 0 & 0 & 1 \end{bmatrix}$$

$$\xrightarrow[R_3=R_3-3\times R_1]{R_2=R_2-2\times R_1} \begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 0 & 0 & -1 & | & -2 & 1 & 0 \\ 0 & -1 & -3 & | & -3 & 0 & 1 \end{bmatrix}$$

$$\xrightarrow{R_2 \rightleftharpoons R_3} \begin{bmatrix} 1 & 2 & 3 & | & 1 & 0 & 0 \\ 0 & -1 & -3 & | & -3 & 0 & 1 \\ 0 & 0 & -1 & | & -2 & 1 & 0 \end{bmatrix}$$

$$\xrightarrow[R_2=-R_2, R_3=-R_3]{R_1=R_1+2\times R_2} \begin{bmatrix} 1 & 0 & -3 & | & -5 & 0 & 2 \\ 0 & 1 & 3 & | & 3 & 0 & -1 \\ 0 & 0 & 1 & | & 2 & -1 & 0 \end{bmatrix}$$

$$\xrightarrow[R_2=R_2+3\times R_3]{R_1=R_1-3\times R_3} \begin{bmatrix} 1 & 0 & 0 & | & 1 & -3 & 2 \\ 0 & 1 & 0 & | & -3 & 3 & -1 \\ 0 & 0 & -1 & | & -2 & 1 & 0 \end{bmatrix}$$

$$\xrightarrow{R_3=-R_3} \begin{bmatrix} 1 & 0 & 0 & | & 1 & -3 & 2 \\ 0 & 1 & 0 & | & -3 & 3 & -1 \\ 0 & 0 & 1 & | & 2 & -1 & 0 \end{bmatrix}$$

$$\mathbf{A}^{-1} = \begin{bmatrix} 1 & -3 & 2 \\ -3 & 3 & -1 \\ 2 & -1 & 0 \end{bmatrix}$$

$$\mathbf{A}^{-1}\mathbf{b} = \begin{bmatrix} 1 & -3 & 2 \\ -3 & 3 & -1 \\ 2 & -1 & 0 \end{bmatrix} \times \begin{bmatrix} -1 \\ 3 \\ 8 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \times (-1) + (-3) \times 3 + 2 \times 8 \\ -3 \times (-1) + 3 \times 3 + (-1) \times 8 \\ 2 \times (-1) + (-1) \times 3 + 0 \times 8 \end{bmatrix}$$

$$= \begin{bmatrix} 6 \\ 4 \\ -5 \end{bmatrix}$$

[9]:
```
print('A_inverse = \n', np.linalg.inv(A))
```

```
A_inverse =
 [[ 1. -3.  2.]
 [-3.  3. -1.]
 [ 2. -1.  0.]]
```

[10]:
```
print('(A_inverse)(b) = \n', np.matmul(np.linalg.inv(A), b))
```

```
(A_inverse)(b) =
 [[ 6.  4. -5.]]
```

13.

$$\mathbf{A} \circ \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \circ \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$$
$$= \begin{bmatrix} 1 \times 1 & 2 \times 2 & 3 \times 3 \\ 2 \times 2 & 4 \times 4 & 5 \times 5 \\ 3 \times 3 & 5 \times 5 & 6 \times 6 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 4 & 9 \\ 4 & 16 & 25 \\ 9 & 25 & 36 \end{bmatrix}$$

14.

$$\mathbf{b} \circ \mathbf{c} = \begin{bmatrix} -1 \\ 3 \\ 8 \end{bmatrix} \circ \begin{bmatrix} 4 \\ -3 \\ 6 \end{bmatrix} = \begin{bmatrix} -1 \times 4 \\ 3 \times (-3) \\ 8 \times 6 \end{bmatrix}$$
$$= \begin{bmatrix} -4 \\ -9 \\ 48 \end{bmatrix}$$

3.2 7

[**8 points**] **Eigenvectors and eigenvalues**. Eigenvectors and eigenvalues are useful for some machine learning algorithms, but the concepts take time to solidly grasp. For an intuitive review of these concepts, explore this interactive website at Setosa.io. Also, the series of linear algebra videos by Grant Sanderson of 3Brown1Blue are excellent and can be viewed on youtube here.

1. Calculate the eigenvalues and corresponding eigenvectors of matrix $\mathbf{A}$ above, from the last question.
2. Choose one of the eigenvector/eigenvalue pairs, $\mathbf{v}$ and λ, and show that $\mathbf{Av} = \lambda\mathbf{v}$. Also show that this relationship extends to higher orders: $\mathbf{AAv} = \lambda^2\mathbf{v}$
3. Show that the eigenvectors are orthogonal to one another (e.g. their inner product is zero). This is true for real, symmetric matrices.

ANSWER

1.Calculate the eigenvalues and corresponding eigenvectors of matrix   above, from the last question.

$$
\begin{aligned}
&det[\mathbf{A} - \lambda\mathbf{I}] \\
&= (1-\lambda)det\begin{bmatrix}4-\lambda & 5 \\ 5 & 6-\lambda\end{bmatrix} - 2det\begin{bmatrix}2 & 5 \\ 3 & 6-\lambda\end{bmatrix} + 3det\begin{bmatrix}2 & 4-\lambda \\ 3 & 5\end{bmatrix} \\
&= (1-\lambda)((4-\lambda)(6-\lambda)-25) - 2(2(6-\lambda)-15) + 3(10-3(4-\lambda)) \\
&= -\lambda^3 + 11\lambda^2 + 4\lambda - 1 \\
&= 0
\end{aligned}
$$

$$
\begin{aligned}
\lambda_1 &= 11.34481428 \\
\lambda_2 &= -0.51572947 \\
\lambda_3 &= 0.17091519
\end{aligned}
$$

$$v_1 = \begin{bmatrix}-0.32798528 \\ -0.59100905 \\ -0.73697623\end{bmatrix}$$

$$v_2 = \begin{bmatrix}-0.73697623 \\ -0.32798528 \\ 0.59100905\end{bmatrix}$$

$$v_3 = \begin{bmatrix}0.59100905 \\ -0.73697623 \\ 0.32798528\end{bmatrix}$$

[11]:
```python
import numpy as np
from numpy import linalg as LA
```

[12]:
```python
A = [[1, 2, 3],
    [2, 4, 5],
    [3, 5, 6]]
```

[13]:
```python
A = np.array(A)
```

[14]:
```python
w, v = LA.eig(A)
#numpy.linalg.eig returns a tuple consisting of a vector and an array.
#vector w contains eigenvalues
#array v contains corresponding eigenvectors, one eigenvector per column
```

2.Choose one of the eigenvector/eigenvalue pairs, $\mathbf{v}$ and λ, and show that $\mathbf{Av} = \lambda\mathbf{v}$. Also show that this relationship extends to higher orders: $\mathbf{AAv} = \lambda^2\mathbf{v}$

$$\lambda_1 = 11.34481428$$

$$\mathbf{v_1} = \begin{bmatrix}-0.32798528 \\ -0.59100905 \\ -0.73697623\end{bmatrix}$$

$$\mathbf{Av_1} = \begin{bmatrix}1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6\end{bmatrix}\begin{bmatrix}-0.32798528 \\ -0.59100905 \\ -0.73697623\end{bmatrix} = \begin{bmatrix}-3.72093206 \\ -6.70488789 \\ -8.36085845\end{bmatrix}$$

$$\lambda_1 \mathbf{v_1} = 11.34481428 \begin{bmatrix} -0.32798528 \\ -0.59100905 \\ -0.73697623 \end{bmatrix} = \begin{bmatrix} -3.72093206 \\ -6.70488789 \\ -8.36085845 \end{bmatrix}$$

$$\mathbf{AAv_1} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix} \begin{bmatrix} -0.32798528 \\ -0.59100905 \\ -0.73697623 \end{bmatrix} = \begin{bmatrix} -42.2132832 \\ -76.06570795 \\ -94.85238636 \end{bmatrix}$$

$$\lambda_1{}^2 \mathbf{v_1} = 11.34481428^2 \begin{bmatrix} -0.32798528 \\ -0.59100905 \\ -0.73697623 \end{bmatrix} = \begin{bmatrix} -42.2132832 \\ -76.06570795 \\ -94.85238636 \end{bmatrix}$$

[15]:
```
w1, w2, w3 = w # lambdas
```

[16]:
```
v1, v2, v3= v[:,0], v[:,1], v[:,2] # eigenvectors
```

[17]:
```
print(np.dot(A, v1))
print(w1*v1)
```

```
[-3.72093206 -6.70488789 -8.36085845]
[-3.72093206 -6.70488789 -8.36085845]
```

[18]:
```
print(np.dot(np.dot(A, A), v1))
print(np.dot(np.dot(w1, w1), v1))
```

```
[-42.2132832  -76.06570795 -94.85238636]
[-42.2132832  -76.06570795 -94.85238636]
```

3.Show that the eigenvectors are orthogonal to one another (e.g. their inner product is zero). This is true for real, symmetric matrices.

$$\mathbf{v_1 v_2} = \begin{bmatrix} -0.32798528 & -0.59100905 & -0.73697623 \end{bmatrix} \begin{bmatrix} -0.73697623 \\ -0.32798528 \\ 0.59100905 \end{bmatrix} = 0$$

$$\mathbf{v_1 v_3} = \begin{bmatrix} -0.32798528 & -0.59100905 & -0.73697623 \end{bmatrix} \begin{bmatrix} 0.59100905 \\ -0.73697623 \\ 0.32798528 \end{bmatrix} = 0$$

$$\mathbf{v_2 v_3} = \begin{bmatrix} -0.73697623 & -0.32798528 \; 0.59100905 \end{bmatrix} \begin{bmatrix} 0.59100905 \\ -0.73697623 \\ 0.32798528 \end{bmatrix} = 0$$

[19]:
```
print('v1v2 = ',np.dot(v1,v2))
print('v2v3 = ',np.dot(v2,v3))
print('v1v3 = ',np.dot(v1,v3))
```

```
v1v2 =  -2.220446049250313e-16
v2v3 =  -1.0547118733938987e-15
v1v3 =  -4.440892098500626e-16
```

4 Numerical Programming

4.1 8

[**10 points**] Loading data and gathering insights from a real dataset

Data. The data for this problem can be found in the `data` subfolder in the `assignments` folder on github. The filename is `egrid2016.xlsx`. This dataset is the Environmental Protection Agency's (EPA) Emissions & Generation Resource Integrated Database (eGRID) containing information about all power plants in the United States, the amount of generation they produce, what fuel they use, the location of the plant, and many more quantities. We'll be using a subset of those data.

The fields we'll be using include:

field	description
SEQPLT16	eGRID2016 Plant file sequence number (the index)
PSTATABB	Plant state abbreviation
PNAME	Plant name
LAT	Plant latitude
LON	Plant longitude
PLPRMFL	Plant primary fuel
CAPFAC	Plant capacity factor
NAMEPCAP	Plant nameplate capacity (Megawatts MW)
PLNGENAN	Plant annual net generation (Megawatt-hours MWh)
PLCO2EQA	Plant annual CO2 equivalent emissions (tons)

For more details on the data, you can refer to the eGrid technical documents. For example, you may want to review page 45 and the section "Plant Primary Fuel (PLPRMFL)", which gives the full names of the fuel types including WND for wind, NG for natural gas, BIT for Bituminous coal, etc.

There also are a couple of "gotchas" to watch out for with this dataset: - The headers are on the second row and you'll want to ignore the first row (they're more detailed descriptions of the headers). - NaN values represent blanks in the data. These will appear regularly in real-world data, so getting experience working with it will be important.

Your objective. For this dataset, your goal is answer the following questions about electricity generation in the United States:

(a) Which plant has generated the most energy (measured in MWh)?

(b) What is the name of the northern-most power plant in the United States?

(c) What is the state where the northern-most power plant in the United States is located?

(d) Plot a bar plot showing the amount of energy produced by each fuel for the plant.

(e) From the plot in (d), which fuel for generation produces the most energy (MWh) in the United States?

ANSWER

[20]:
```python
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

[21]:
```python
egrid = pd.read_excel('/Users/yiran/Desktop/IDS705ML/ids705/assignments/data/
↪egrid2016.xlsx',
                      skiprows=[0])
```

[22]:
```python
egrid.head()
```

[22]:
```
   SEQPLT16 PSTATABB                           PNAME        LAT         LON  \
0         1       AK          7-Mile Ridge Wind Project  63.210689 -143.247156
1         2       AK  Agrium Kenai Nitrogen Operations  60.673200 -151.378400
2         3       AK                          Alakanuk  62.683300 -164.654400
3         4       AK               Allison Creek Hydro  61.084444 -146.353333
4         5       AK                            Ambler  67.087980 -157.856719

  PLPRMFL   CAPFAC  NAMEPCAP  PLNGENAN  PLCO2EQA
0     WND      NaN       1.8       NaN       NaN
1      NG      NaN      21.6       NaN       NaN
2     DFO  0.05326       2.6  1213.001  1049.863
3     WAT  0.01547       6.5   881.000     0.000
4     DFO  0.13657       1.1  1315.999  1087.881
```

[23]:
```python
egrid[['LAT','LON','CAPFAC','NAMEPCAP','PLNGENAN','PLCO2EQA']] \
.isnull().sum() / len(egrid)
```

[23]:
```
LAT         0.004223
LON         0.004223
CAPFAC      0.172108
NAMEPCAP    0.001339
PLNGENAN    0.172108
PLCO2EQA    0.210114
dtype: float64
```

(a) Which plant has generated the most energy (measured in MWh)? > Palo Verde

[24]:
```python
egrid['PLNGENAN'].max()
egrid[['PSTATABB','PNAME']][egrid['PLNGENAN']==egrid['PLNGENAN'].max()]
```

[24]:
```
    PSTATABB       PNAME
390       AZ  Palo Verde
```

(b) What is the name of the northern-most power plant in the United States? > Barrow

[25]:
```python
egrid['LAT'].max()
egrid[['PSTATABB','PNAME']][egrid['LAT']==egrid['LAT'].max()]
```

[25]:
```
   PSTATABB   PNAME
11       AK  Barrow
```

(c) What is the state where the northern-most power plant in the United States is located? > Alaska (AK)

[26]:
```python
egrid[egrid['PNAME']=='Barrow']
```

[26]:
```
    SEQPLT16 PSTATABB   PNAME     LAT       LON PLPRMFL   CAPFAC  NAMEPCAP  \
11        12       AK  Barrow  71.292 -156.7786      NG  0.28208      20.3

    PLNGENAN  PLCO2EQA
11   50162.0  44205.17
```

(d) Plot a bar plot showing the amount of energy produced by each fuel for the plant.

[27]:
```python
df = egrid[['PNAME','PLPRMFL']].sort_values(by='PNAME')
df.head()
```

[27]:
```
                        PNAME PLPRMFL
6364  12 Applegate Solar LLC     SUN
3893      126 Grove Solar LLC     SUN
442           1420 Coil Av #C     SUN
6365       145 Talmadge Solar     SUN
3164         1515 S Caron Road      NG
```

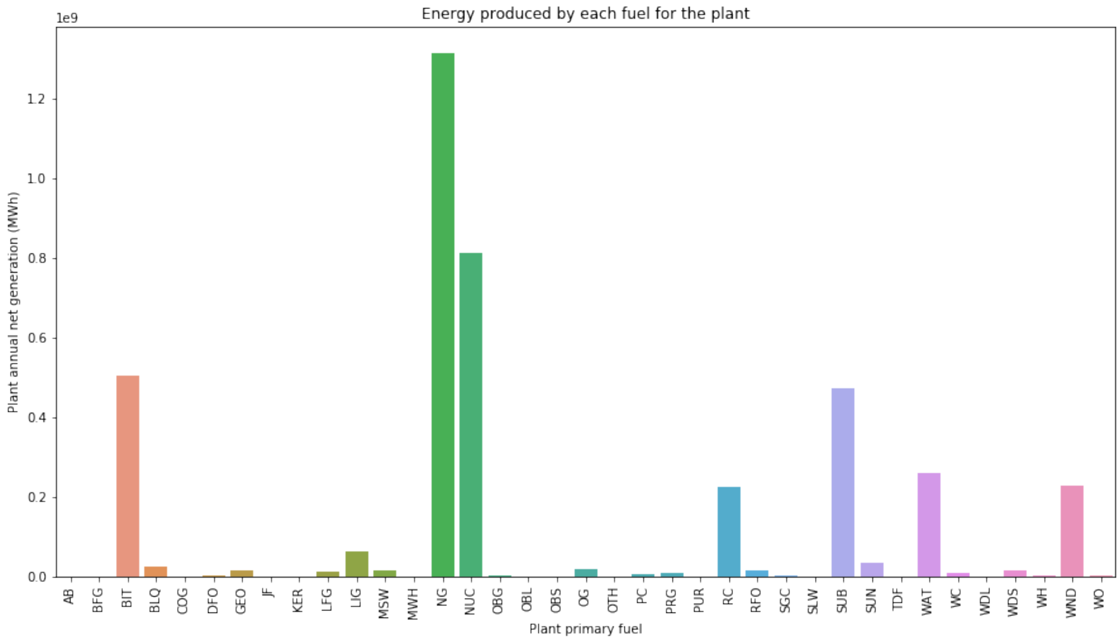
[28]:
```python
# check for one-to-one relationship between columns
def isOneToOne(egrid, col1, col2):
    a = df.groupby(col1)[col2].count().max()
    b = df.groupby(col2)[col1].count().max()
    return a + b == 2

# check whether Plant and Fuel is one-to-one relationship
isOneToOne(egrid, 'PNAME','PLPRMFL')
```

[28]: False

[29]:
```python
egrid_grped = egrid[['PLPRMFL','PLNGENAN']].groupby('PLPRMFL').sum().
 ↪reset_index()
egrid_grped.head()
```

[29]:
```
  PLPRMFL      PLNGENAN
0      AB  9.412637e+05
1     BFG  7.396800e+05
2     BIT  5.049193e+08
3     BLQ  2.609711e+07
4     COG  1.398400e+04
```

[30]:
```
plt.figure(figsize=(15,8))
sns.barplot(x='PLPRMFL',y='PLNGENAN',data=egrid_grped)
plt.xticks(rotation=90)
plt.xlabel('Plant primary fuel')
plt.ylabel('Plant annual net generation (MWh)')
plt.title('Energy produced by each fuel for the plant')
plt.show()
```



(e) From the plot in (e), which fuel for generation produces the most energy (MWh) in the United States? > Natural Gas (NG)

[31]:
```
egrid[['PLPRMFL','PLNGENAN']].groupby('PLPRMFL') \
.sum().sort_values('PLNGENAN', ascending=False).iloc[0]
```

[31]:
```
PLNGENAN    1.314956e+09
Name: NG, dtype: float64
```

4.2 9

[**8 points**] Speed comparison between vectorized and non-vectorized code. Begin by creating an array of 10 million random numbers using the numpy random.randn module. Compute the sum of the squares first in a for loop, then using Numpy's `dot` module. Time how long it takes to compute each and report the results and report the output. How many times faster is the vectorized code than the for loop approach?

*Note: all code should be well commented, properly formatted, and your answers should be output using the `print()` function as follows (where the # represents your answers, to a reasonable precision):

`Time [sec] (non-vectorized): ######`

`Time [sec] (vectorized):     ######`

`The vectorized code is ##### times faster than the vectorized code`

ANSWER

[32]:
```
import numpy as np
import time
```

[33]:
```
np_array = np.random.randn(10000000) # create an array of 10 million random␣
↪numbers
```

[34]:
```
# non-vectorized
sum_of_sqrs = 0
t0 = time.time()
for num in np_array:
    sum_of_sqrs += num**2
t1 = time.time()
time_nonvectorized = t1 - t0
```

[35]:
```
# vectorized
t2 = time.time()
ans = np.dot(np_array, np_array)
t3 = time.time()
time_vectorized = t3 - t2
```

[36]:
```
print('Time [sec] (non-vectorized): %.5f' %time_nonvectorized)
print('Time [sec] (vectorized): %.5f' %time_vectorized)
print('The vectorized code is %.5f times faster than the non-vectorized code'␣
↪%(time_nonvectorized/time_vectorized))
```

```
Time [sec] (non-vectorized): 7.11877
Time [sec] (vectorized): 0.00831
The vectorized code is 856.74075 times faster than the non-vectorized code
```

4.3 10

[**10 points**] One popular Agile development framework is Scrum (a paradigm recommended for data science projects). It emphasizes the continual evolution of code for projects, becoming progressively better, but starting with a quickly developed minimum viable product. This often means that code written early on is not optimized, and that's a good thing - it's best to get it to work first before optimizing. Imagine that you wrote the following code during a sprint towards getting an end-to-end system working. Vectorize the following code and show the difference in speed between the current implementation and a vectorized version.

The function below computes the function $f(x, y) = x^2 - 2y^2$ and determines whether this quantity is above or below a given threshold, `thresh=0`. This is done for $x, y \in \{-4, 4\}$, over a 2,000-by-2,000 grid covering that domain.

(a) Vectorize this code and demonstrate (as in the last exercise) the speed increase through vectorization and (b) plot the resulting data - both the function $f(x, y)$ and the thresholded output - using `imshow` from `matplotlib`.

Hint: look at the `numpy` `meshgrid` documentation

ANSWER

[37]:
```
import numpy as np
import time
import matplotlib.pyplot as plt
from matplotlib import colors
```

[38]:
```
nvalues = 2000
xvalues = np.linspace(-4,4,nvalues)
yvalues = np.linspace(-4,4,nvalues)
thresh  = 0
```

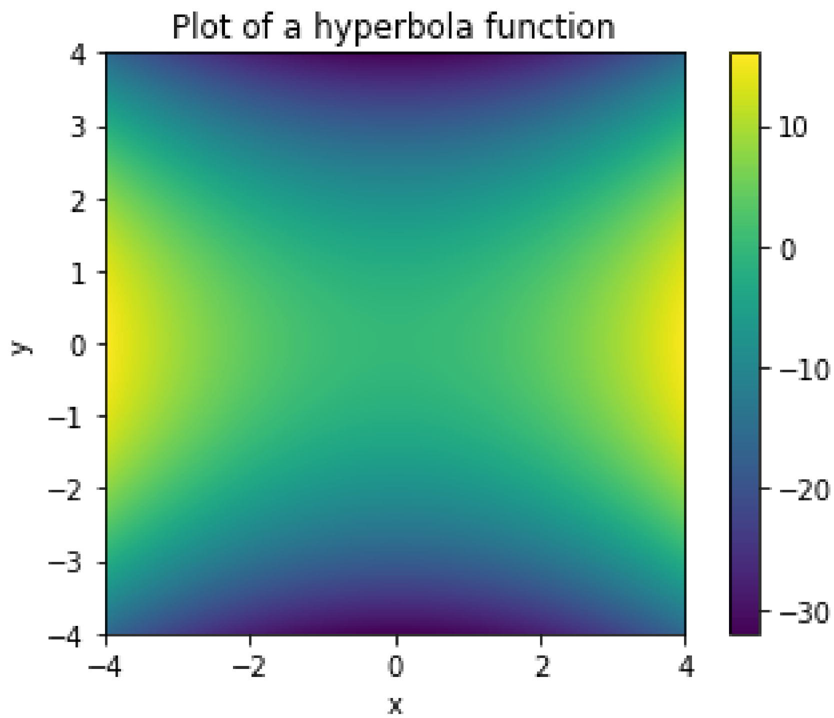
[39]:
```
# Nonvectorized implementation
t0 = time.time()
f = np.zeros((nvalues,nvalues))
f_thresholded = np.zeros((nvalues,nvalues))
for ix, x in enumerate(xvalues):
    for iy, y in enumerate(yvalues):
        f[ix,iy]             = x**2 - 2 * y**2
        f_thresholded[ix,iy] = f[ix,iy] > thresh
t1 = time.time()
time_nonvectorized = t1 - t0
```

[40]:
```
# Vectorized implementation
t2 = time.time()
X, Y = np.meshgrid(xvalues, yvalues)
f = X ** 2 - 2 * Y**2
f_thresholded = f > thresh
t3 = time.time()
time_vectorized = t3 - t2
```

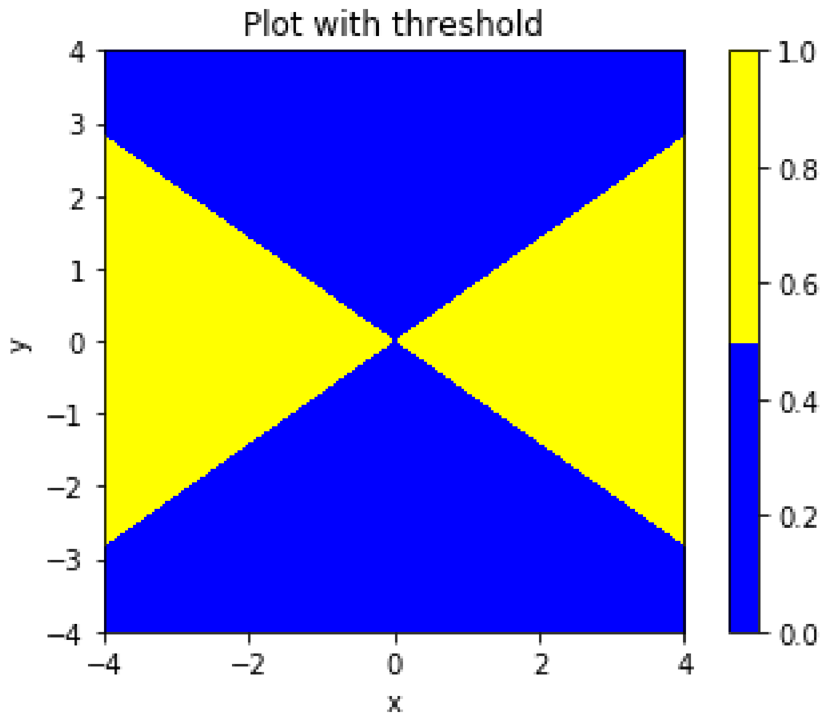
[41]:
```
# Vectorization speed performance results
print('Time [sec] (non-vectorized): %.5f' %time_nonvectorized)
print('Time [sec] (vectorized): %.5f' %time_vectorized)
print('The vectorized code is %.5f times faster than the non-vectorized code'␣
 ↪%(time_nonvectorized/time_vectorized))
```

```
Time [sec] (non-vectorized): 10.16485
Time [sec] (vectorized): 0.12985
The vectorized code is 78.28385 times faster than the non-vectorized code
```

[42]:
```
# Plot the result for function
plt.imshow(f, extent = [-4, 4, -4, 4])
plt.xlabel('x')
plt.ylabel('y')
plt.title('Plot of a hyperbola function')
plt.colorbar()
plt.show()
```



[43]:
```
# Plot the result for threshold
f_thres = f.copy()
f_thres[f_thres > thresh] = 1
f_thres[f_thres < thresh] = 0

cmap = colors.ListedColormap(['blue','yellow'])
plt.imshow(f_thres, cmap=cmap, extent = [-4, 4, -4, 4])
plt.colorbar()
plt.xlabel('x')
plt.ylabel('y')
plt.title('Plot with threshold')
plt.show()
```



4.4 11

[10 points] This exercise will walk through some basic numerical programming exercises. 1. Synthesize $n = 10^4$ normally distributed data points with mean $\mu = 2$ and a standard deviation of $\sigma = 1$. Call these observations from a random variable X, and call the vector of observations that you generate, $\mathbf{x}$. 2. Calculate the mean and standard deviation of $\mathbf{x}$ to validate (1) and provide the result to a precision of four significant figures. 3. Plot a histogram of the data in $\mathbf{x}$ with 30 bins 4. What is the 90th percentile of $\mathbf{x}$? The 90th percentile is the value below which 90% of observations can be found. 5. What is the 99th percentile of $\mathbf{x}$? 6. Now synthesize $n = 10^4$ normally distributed data points with mean $\mu = 0$ and a standard deviation of $\sigma = 3$. Call these observations from a random variable Y, and call the vector of observations that you generate, $\mathbf{y}$. 7. Create a new figure and plot the histogram of the data in $\mathbf{y}$ on the same axes with the histogram of $\mathbf{x}$, so that both histograms can be seen and compared. 8. Using the observations from $\mathbf{x}$ and $\mathbf{y}$, estimate $E[XY]$

ANSWER

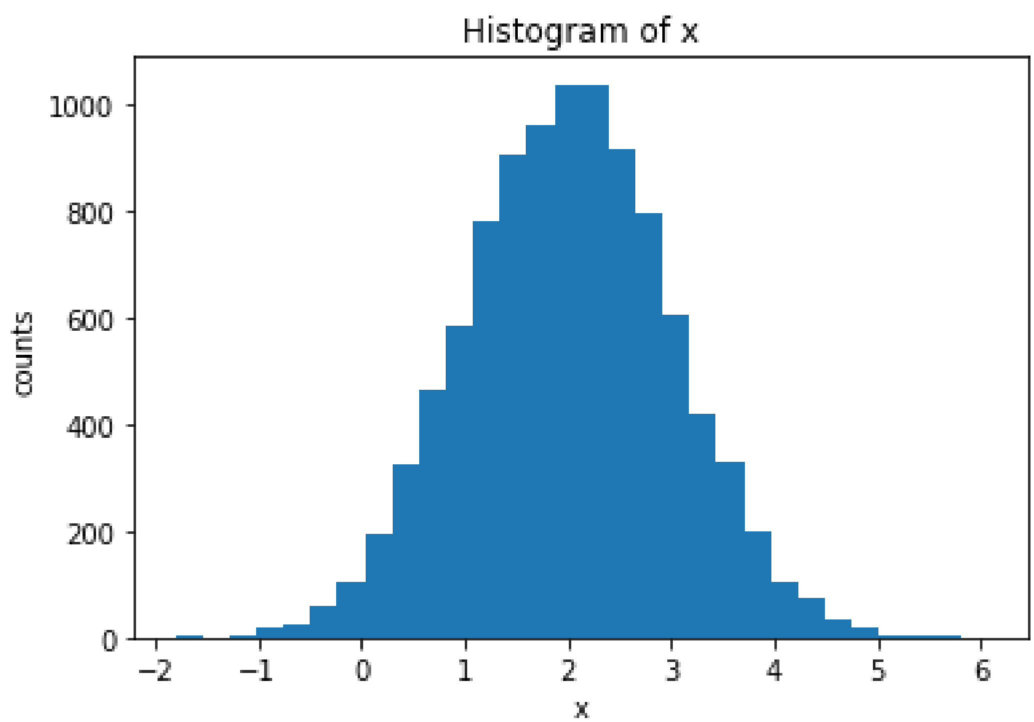
[44]:
```
import numpy as np
import matplotlib.pyplot as plt
np.random.seed(123)
```

[45]:
```
# 1
mu, sigma = 2, 1
x = np.random.normal(mu, sigma, 10000)
```

[46]:
```python
# 2
print('mean of x: %.3f' %x.mean())
print('standard deviation of x: %.4g' %x.std())
```

```
mean of x: 2.010
standard deviation of x: 0.9981
```

[47]:
```python
# 3
plt.hist(x, bins=30)
plt.xlabel('x')
plt.ylabel('counts')
plt.title('Histogram of x')
plt.show()
```



[48]:
```python
# 4
print('The 90th percentile of x is %.4g' %np.percentile(x, 90))
```

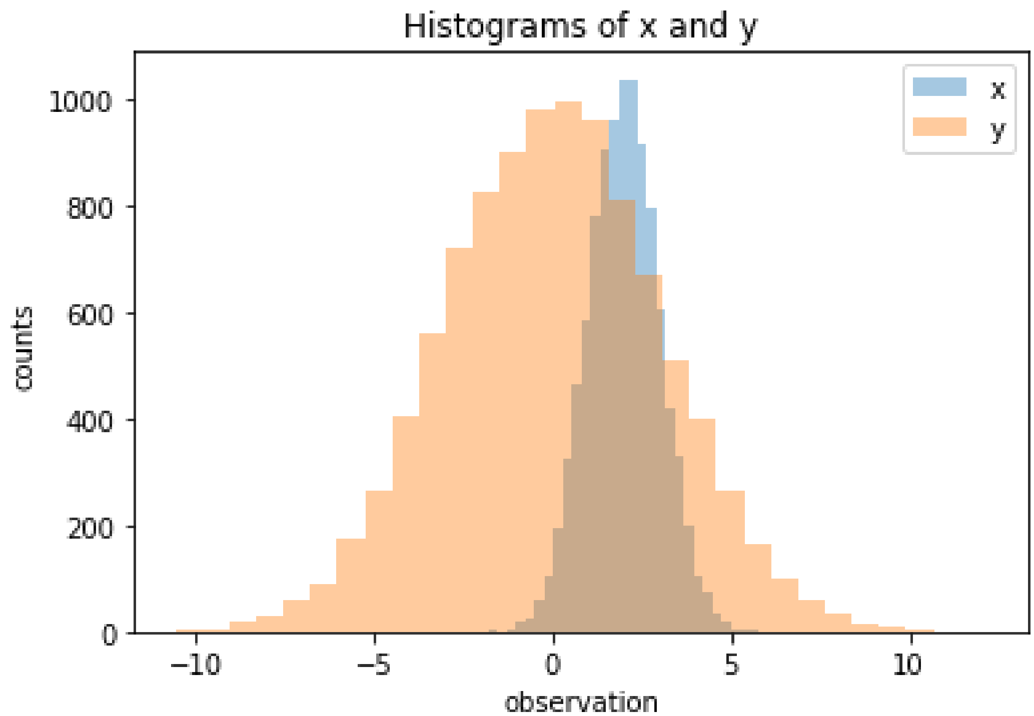
```
The 90th percentile of x is 3.289
```

[49]:
```python
# 5
print('The 99th percentile of x is %.4g' %np.percentile(x, 99))
```

```
The 99th percentile of x is 4.326
```

[50]:
```
# 6
mu, sigma = 0, 3
y = np.random.normal(mu, sigma, 10000)
```

[51]:
```
# 7
plt.hist(x, bins=30, alpha=0.4, label='x')
plt.hist(y, bins=30, alpha=0.4, label='y')
plt.title('Histograms of x and y')
plt.xlabel('observation')
plt.ylabel('counts')
plt.legend(loc='upper right')
plt.show()
```



[52]:
```
# 8
print('E[XY] = %.5f' %np.mean(x*y))
```

```
E[XY] = 0.07897
```

5 Version Control via Git

5.1 12

[**1 point**] You will need to use Git to submit assignments and in the course projects and is generally a version control and collaboration tool. You can even use some Git repositories (e.g. Github) as hosts for website, such as with the course website.

Complete the Atlassian Git tutorial, specifically the following listed sections. Try each concept that's presented. For this tutorial, instead of using BitBucket as your remote repository host, you may use your preferred platform such as Github or Duke's Gitlab. 1. What is version control 2. What is Git 3. Install Git 4. Setting up a repository 5. Saving changes 6. Inspecting a repository 7. Undoing changes 8. Rewriting history 9. Syncing 10. Making a pull request 11. Using branches 12. Comparing workflows

I also have created two videos on the topic to help you understand some of these concepts: Git basics and a step-by-step tutorial.

For your answer, affirm that you *either* completed the tutorial or have previous experience with all of the concepts above. Do this by typing your name below and selecting the situation that applies from the two options in brackets.

ANSWER

I, Yiran, affirm that I have completed the above tutorial.

6 Exploratory Data Analysis

6.1 13

[**20 points**] Here you'll bring together some of the individual skills that you demonstrated above and create a Jupyter notebook based blog post on data analysis.

1. Find a dataset that interests you and relates to a question or problem that you find intriguing
2. Using a Jupyter notebook, describe the dataset, the source of the data, and the reason the dataset was of interest.
3. Check the data and see if they need to be cleaned: are there missing values? Are there clearly erroneous values? Do two tables need to be merged together? Clean the data so it can be visualized.
4. Plot the data, demonstrating interesting features that you discover. Are there any relationships between variables that were surprising or patterns that emerged? Please exercise creativity and curiosity in your plots.
5. What insights are you able to take away from exploring the data? Is there a reason why analyzing the dataset you chose is particularly interesting or important? Summarize this as if your target audience was the readership of a major news organization - boil down your findings in a way that is accessible, but still accurate.

Here your analysis will evaluated based on: 1. Data cleaning: did you look for and work to resolve issues in the data? 2. Quality of data exploration: did you provide plots demonstrating interesting aspects of the data? 3. Interpretation: Did you clearly explain your insights? Restating the data, alone, is not interpretation. 5. Professionalism: Was this work done in a way that exhibits professionalism through clarity, organization, high quality figures and plots, and meaningful descriptions?

ANSWER

6.2 Heart Disease

The data is taken from UCI Machine Learning Repository with experiment data in the Cleveland database. The database concentrated on attempting to distinguish presence of heart disease from absence. The intention is to find any trends in heart data to better find or predict any abnormal events of heart health.

6.2.1 Features

- age (age in years)
- sex
  - 1 = male
  - 0 = female
- cp (chest pain type)
  - 0 = typical angina
  - 1 = atypical angina
  - 2 = non-anginal pain
  - 3 = asymptomatic
- trestbps (resting blood pressure, in mm Hg on admission to the hospital)
- chol (serum cholestoral in mg/dl)
- fbs (fasting blood sugar > 120 mg/dl)
  - 1 = true
  - 0 = false
- restecg (resting electrocardiographic results)
  - 0 = showing probable or definite left ventricular hypertrophy by Estes' criteria
  - 1 = normal
  - 2 = having ST-T wave abnormality
- thalach (maximum heart rate achieved)
- exang (exercise induced angina)
  - 1 = yes
  - 0 = no
- oldpeak (ST depression induced by exercise relative to rest, 'ST' relates to positions on the ECG plot)
- slope (the slope of the peak exercise ST segment)
  - 0 = downsloping
  - 1 = flat
  - 2 = upsloping
- ca (number of major vessels (0-3) colored by flourosopy)
- thal (a blood disorder called thalassemia)
  - 1 = fixed defect
  - 2 = normal
  - 3 = reversable defect
- target (heart disease)
  - 1 = no, no disease
  - 0 = yes, have disease

[53]:
```
import os
import numpy as np
```

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# loading data
ht = pd.read_csv('/Users/yiran/Desktop/IDS705ML/ids705/assignments/heart.csv')
ht.head()
```

[53]:

```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   3       145   233    1        0      150      0      2.3      0
1   37    1   2       130   250    0        1      187      0      3.5      0
2   41    0   1       130   204    0        0      172      0      1.4      2
3   56    1   1       120   236    0        1      178      0      0.8      2
4   57    0   0       120   354    0        1      163      1      0.6      2

   ca  thal  target
0   0     1       1
1   0     2       1
2   0     2       1
3   0     2       1
4   0     2       1
```

[54]:

```python
# change outcome variable *target* to 1=disease and 0=no disease
ht.target[ht.target==0]='yes'
ht.target[ht.target==1]='no'
```

```
/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[55]:

```python
# rename numbers in categorical variables to string for better intepretation
ht.cp[ht.cp==0]='typical angina'
ht.cp[ht.cp==1]='atypical angina'
ht.cp[ht.cp==2]='non-anginal pain'
ht.cp[ht.cp==3]='asymptomatic'

ht.restecg[ht.restecg==0]='left ventricular hypertrophy'
ht.restecg[ht.restecg==1]='normal'
ht.restecg[ht.restecg==2]='ST-T wave abnormality'

ht.thal[ht.thal==1]='fixed defect'
ht.thal[ht.thal==2]='normal'
ht.thal[ht.thal==3]='reversible defect'
```

```python
ht.slope[ht.slope==0]='downsloping'
ht.slope[ht.slope==1]='flat'
ht.slope[ht.slope==2]='upsloping'
```

```
/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:7:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  import sys
/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:8:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:9:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  if __name__ == '__main__':
/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:11:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  # This is added back by InteractiveShellApp.init_path()
/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:15:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  from ipykernel import kernelapp as app
```

```
/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:16:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  app.launch_new_instance()
/Users/yiran/anaconda2/lib/python3.7/site-packages/ipykernel_launcher.py:17:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

[56]:
```
ht.info() # no missing value
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age         303 non-null int64
sex         303 non-null int64
cp          303 non-null object
trestbps    303 non-null int64
chol        303 non-null int64
fbs         303 non-null int64
restecg     303 non-null object
thalach     303 non-null int64
exang       303 non-null int64
oldpeak     303 non-null float64
slope       303 non-null object
ca          303 non-null int64
thal        303 non-null object
target      303 non-null object
dtypes: float64(1), int64(8), object(5)
memory usage: 33.3+ KB
```

[57]:
```
# remove two 0 values in thal (these were NA values in the original dataset)
ht = ht[ht.thal!=0]
```

[58]:
```
ht.sample(5)
```

[58]:

	age	sex	cp	trestbps	chol	fbs	\
136	60	0	non-anginal pain	120	178	1	
172	58	1	atypical angina	120	284	0	
185	44	1	typical angina	112	290	0	
72	29	1	atypical angina	130	204	0	
82	60	0	non-anginal pain	102	318	0	

```
                        restecg  thalach  exang  oldpeak      slope  ca  \
136                      normal       96      0      0.0  upsloping   0
172  left ventricular hypertrophy      160      0      1.8       flat   0
185  left ventricular hypertrophy      153      0      0.0  upsloping   1
72   left ventricular hypertrophy      202      0      0.0  upsloping   0
82                       normal      160      0      0.0  upsloping   1

       thal target
136  normal     no
172  normal    yes
185  normal    yes
72   normal     no
82   normal     no
```

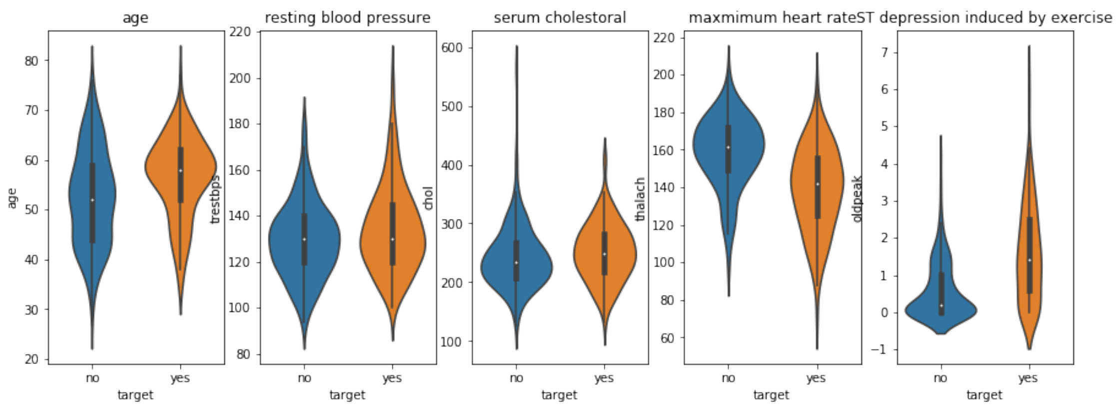
6.2.2 Exploratory Data Analysis

It is known that there are several heart disease risk factors including high cholesterol, high blood pressure, diabetes, etc., which can be resulted from unhealthy diet or behaviors such as smoking, drinking, or physical inactivity. Inherently, factors including age, gender, and blood disease can affect heart conditions as well. These factors might be of importance for the predictability of heart diseases.

[59]:
```python
# overview paired plots
# sns.pairplot(ht)
# plt.show()
```

[60]:
```python
## 5 continuous variables
plt.figure(figsize=(15,5))
# age
plt.subplot(1, 5, 1)
sns.violinplot(x=ht.target, y=ht.age)
plt.title("age")

# trestbps
plt.subplot(1, 5, 2)
sns.violinplot(x=ht.target, y=ht.trestbps)
plt.title('resting blood pressure')

# chol
plt.subplot(1, 5, 3)
sns.violinplot(x=ht.target, y=ht.chol)
plt.title('serum cholestoral')

# thalach
plt.subplot(1, 5, 4)
sns.violinplot(x=ht.target, y=ht.thalach)
```

```
plt.title('maxmimum heart rate')

# oldpeak
plt.subplot(1, 5, 5)
sns.violinplot(x=ht.target, y=ht.oldpeak)
plt.title('ST depression induced by exercise')

plt.show()
```



From the violin plots of the five continuous variables, it could be seen that people suffering from heart diseases are mostly concentrated from age 55 to 65. Patients aging from 20-30 are much less likely to suffer from heart diseases. As age increases, the likelihood of having heart diseases increases.

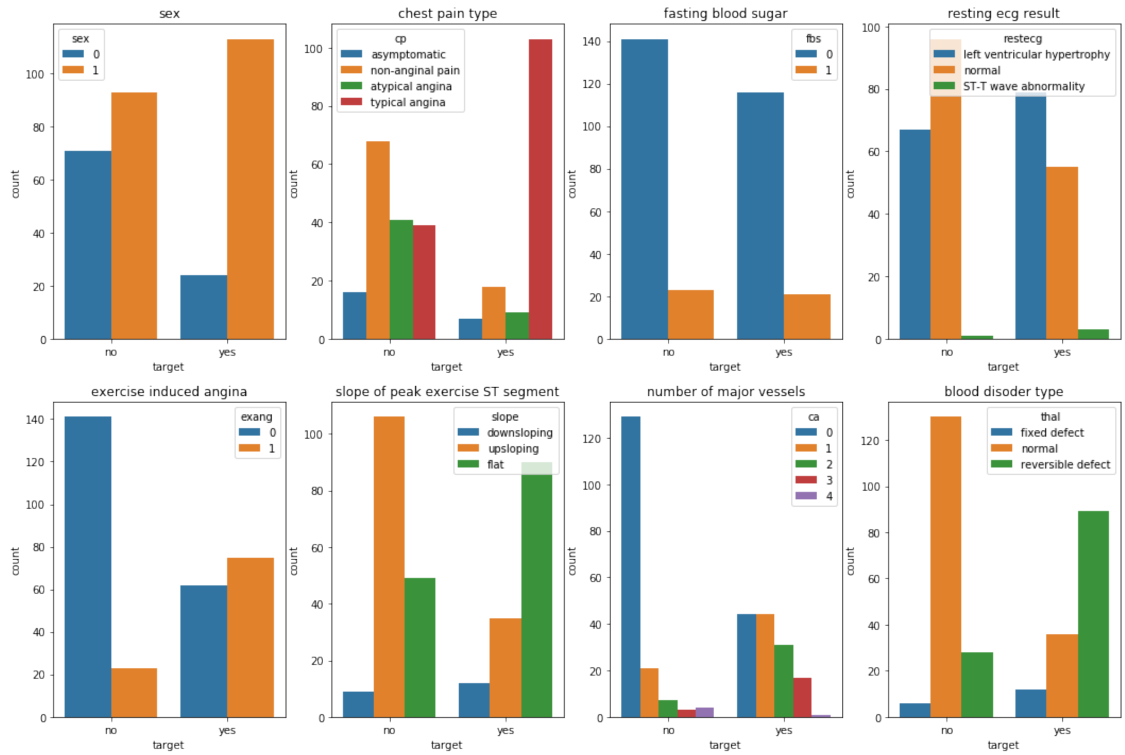
Patients who are more likely to suffer from the heart disease have slightly higher resting blood pressure and serum cholesterol. This agrees with our initial hypothesis that high blood pressure and high cholesterol level are more likely to induce heart diseases.

ST depression refers to a finding on an electrocardiogram (ECG), wherein the trace in the ST segment is abnormally low below the baseline. Having a much higher number of ST depression induced by exercises is an indication of heart may not be normal functioning under exercise relative to rest. In comparison, it can be seen that most patients without heart diseases have a mean value around 0 of ST depression induced by exercise.

According to research, the risk of having heart diseases will be reduced with a low resting heart rate and high maximum heart rate. Such notion is also supported from observations here that patients with heart diseases tend to have a lower maximum heart rate than those without.

[61]:
```
## 8 categorical variables
plt.figure(figsize=(18,12))
# sex
plt.subplot(2,4,1)
sns.countplot('target',hue='sex',data=ht)
plt.title('sex')
```

```python
# cp
plt.subplot(2,4,2)
sns.countplot('target',hue='cp',data=ht)
plt.title('chest pain type')

# fbs
plt.subplot(2,4,3)
sns.countplot('target',hue='fbs',data=ht)
plt.title('fasting blood sugar')

#restecg
plt.subplot(2,4,4)
sns.countplot('target',hue='restecg',data=ht)
plt.title('resting ecg result')

# exang
plt.subplot(2,4,5)
sns.countplot('target',hue='exang',data=ht)
plt.title('exercise induced angina')

# slope
plt.subplot(2,4,6)
sns.countplot('target',hue='slope',data=ht)
plt.title('slope of peak exercise ST segment')

# ca
plt.subplot(2,4,7)
sns.countplot('target',hue='ca',data=ht)
plt.title('number of major vessels')

# thal
plt.subplot(2,4,8)
sns.countplot('target',hue='thal',data=ht)
plt.title('blood disoder type')

plt.show()
```



It could be seen from the plots that men have a higher chance of heart disease compared to women. Such observation agrees with the general claim in the medical field that heart attacks are more common in men than in women.

Regarding chest pain types, typical angina is substernal chest pain or discomfort provoked by exertion or emotional stress and relieved by rest or nitroglycerin. It can be seen that having typical angina is much more likely to exhibit heart diseases. On the other hand, with symptoms of non-anginal pain or atypical angina, there might be many other factors involved encompassing possibilities besides heart diseases. Relatively speaking, patients who are experiencing asymptomatic, non-anginal pain or atypical angina might be less likely to have heart diseases.

There seems to be an equal split for patients having fasting blood sugar over 120 mg/dl among with heart diseases and without group. However, it should be noted that for patients whose fasting blood sugar is less than 120 mg/dl, there is a much higher number of them not having heart diseases. This suggests that in order to prevent possibility of heart diseases, it is better to have a relatively lower fasting blood sugar to mitigate the risks.

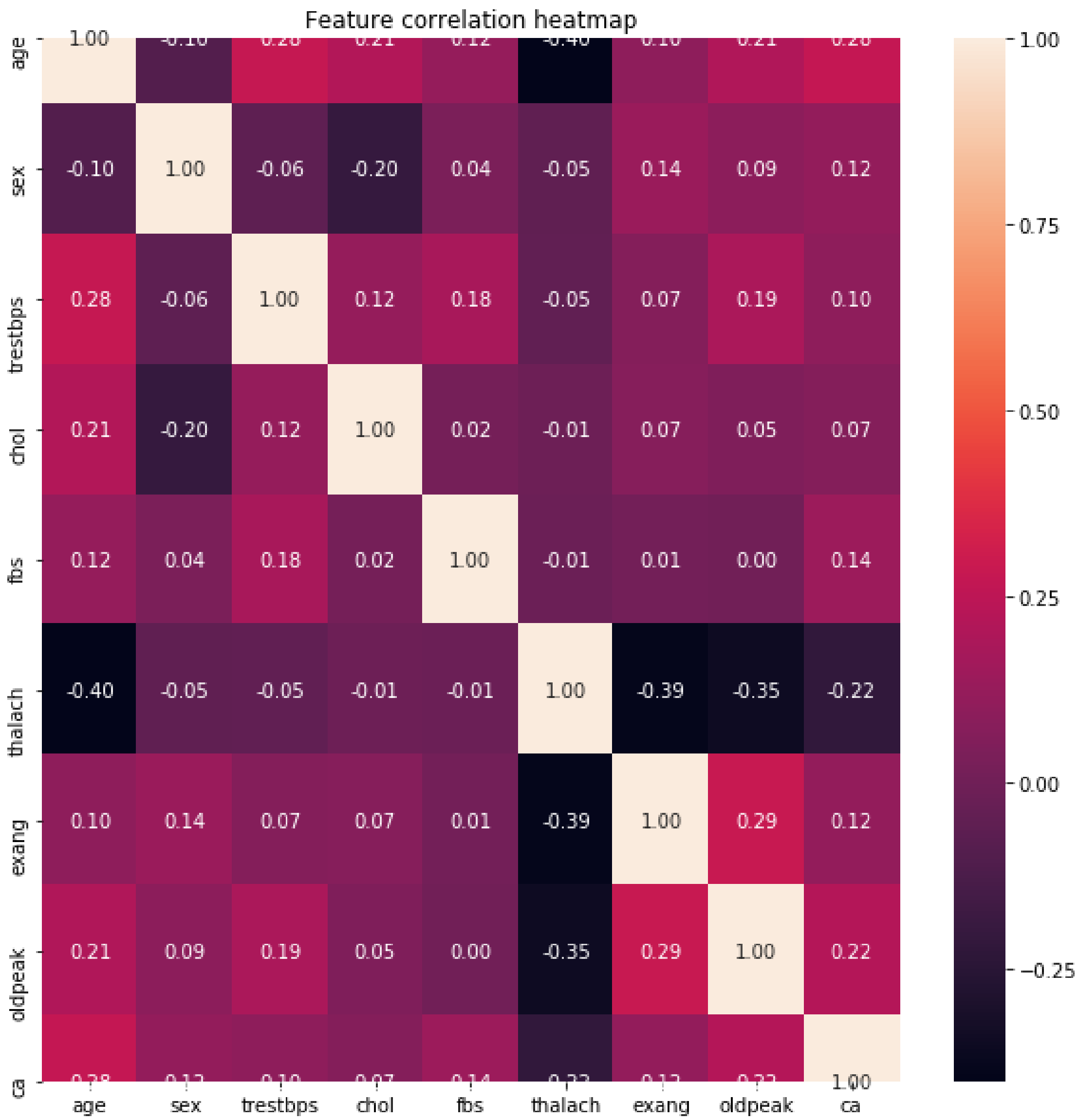
From the resting ECG results, patients who have left ventricular hypertrophy or ST-T wave abnormality tend to have a higher chance of having heart diseases. Patients who have exercise-induced angina also have higher risk of having heart diseases. It would be a good idea to do regular checks for early detection of possible heart conditions.

The ST segment represents the heart's electrical activity immediately after ventricles contracted. The shape and direction of the ST segment are very important where upward or downward shifts can represent decreased blood flow to the heart from a variety of causes. It seems both downsloping and flat slope of peak exercise ST segment are bad signs with higher occurrence of heart diseases, while there are less patients with heart diseases who have upsloping pattern in exercises.

As heart disease is highly concerned with blood flow, intuitively one would think that the more major vessels one has, the less likely for the patients to have heart diseases. However, the plot shows increase chance of having heart diseases as the number of major vessels increases except for the few patients who have four major vessels. This might be of interest to see where these major vessels located, individual conditions and impacts on the heart.

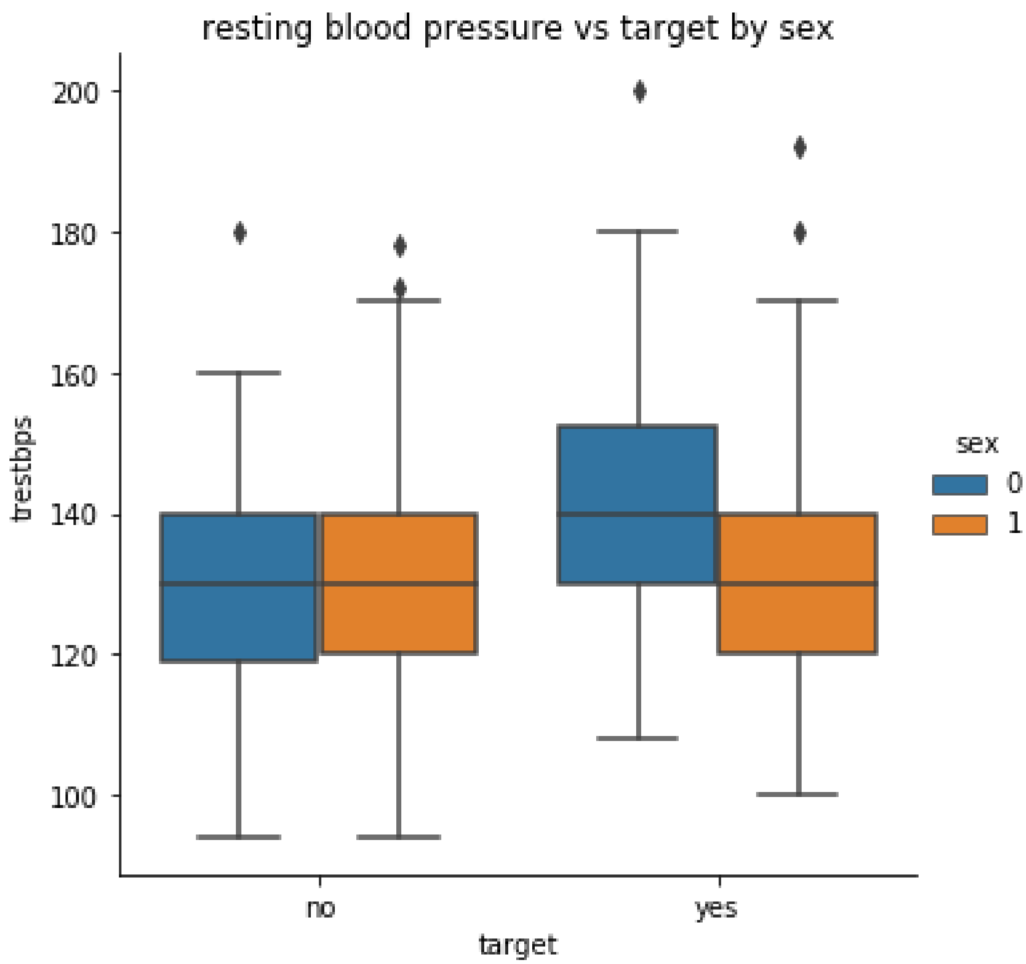
The blood disorder (thalassemia) causes the body to have less hemoglobin than normal which enables red blood cells to carry oxygen. Patients with thalassemia of reversible defect have a significantly larger number of heart disease occurrence, and those who have thalassemia of fixed effect also have higher heart disease occurrence but to a smaller magnitude.

[62]:
```
## correlations
plt.figure(figsize=(10,10))
sns.heatmap(ht.corr(),fmt='.2f', annot=True)
plt.title('Feature correlation heatmap')
plt.show()
```

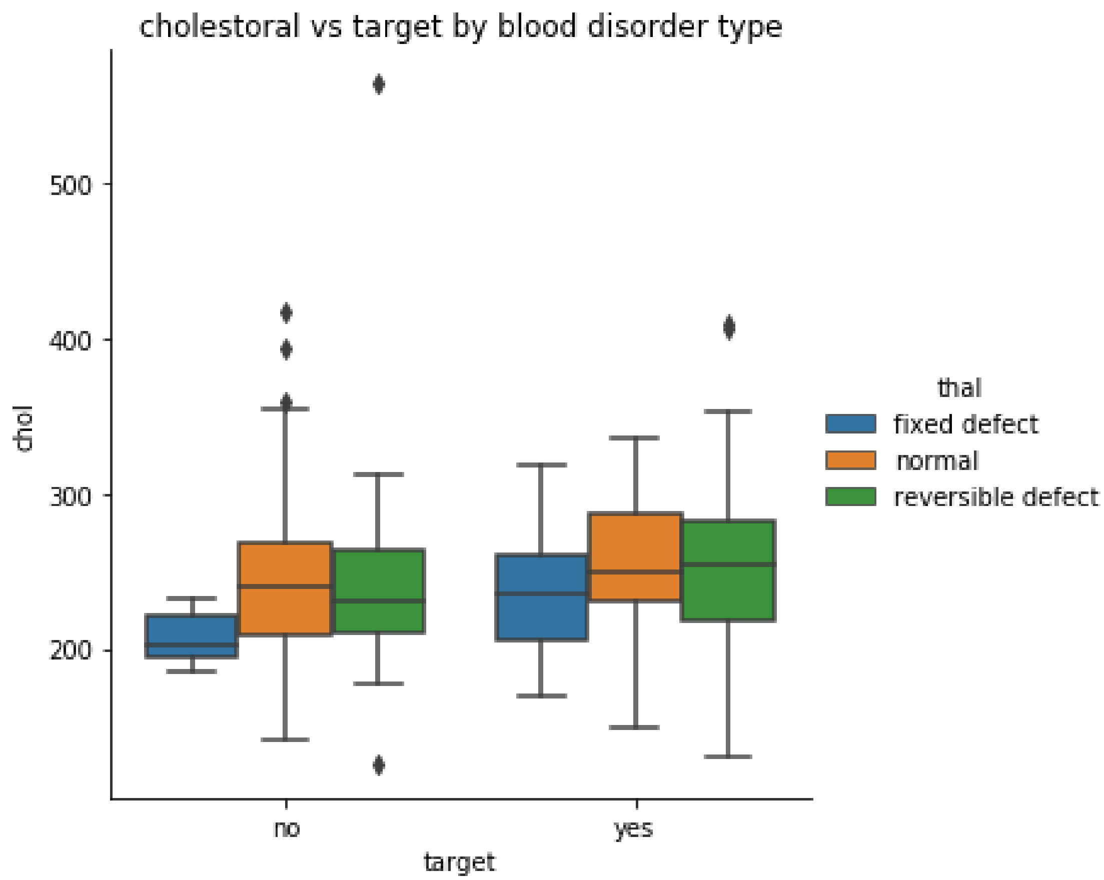


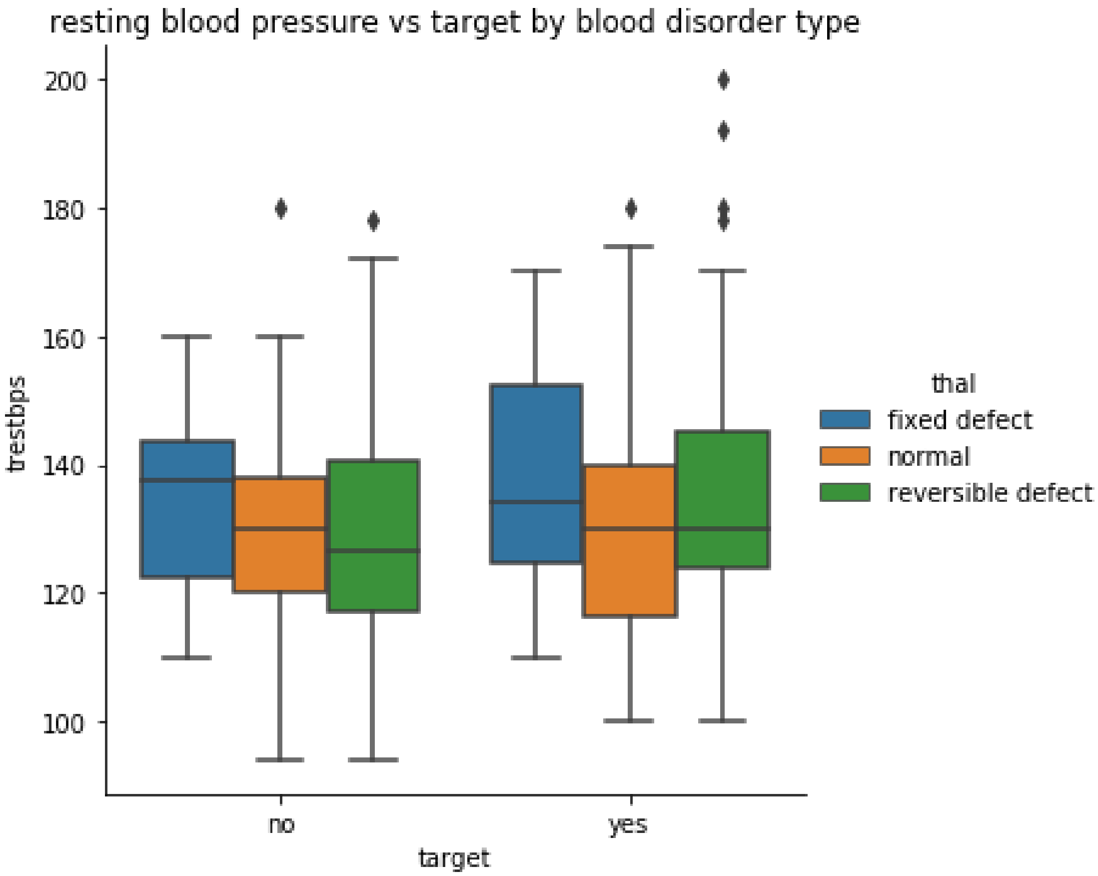
[63]:
```
## interactions
# sex and tresbps
sns.catplot(x='target',y='trestbps',hue='sex',kind='box', data=ht)
plt.title('resting blood pressure vs target by sex')

# thal and chol
sns.catplot(x='target',y='chol',hue='thal',kind='box', data=ht)
plt.title('cholestoral vs target by blood disorder type')

# thal and trestbps
sns.catplot(x='target',y='trestbps',hue='thal',kind='box', data=ht)
plt.title('resting blood pressure vs target by blood disorder type')
```

```
plt.show()
```

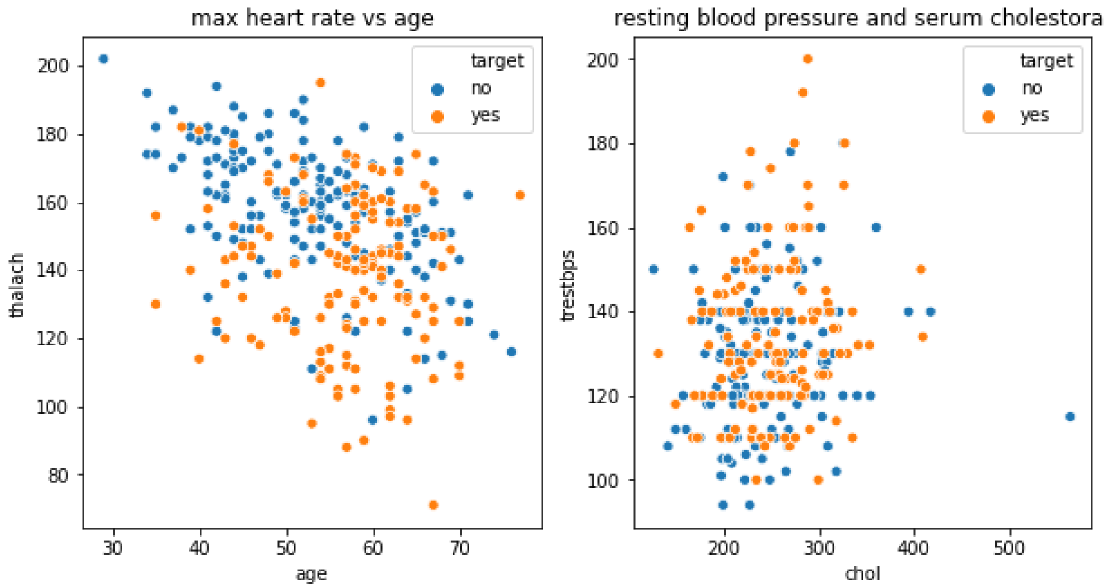


resting blood pressure vs target by sex
trestbps
200
180
160
140
120
100
no
yes
target
sex
0
1


cholestoral vs target by blood disorder type
500
400
300
200
chol
no
yes
target
thal
fixed defect
normal
reversible defect



[64]:
```
plt.figure(figsize=(10,5))
# thalach and age
plt.subplot(1,2,1)
sns.scatterplot(x="age", y="thalach",data=ht,hue="target")
plt.title('max heart rate vs age')

# tresbps and chol
plt.subplot(1,2,2)
sns.scatterplot(x="chol", y="trestbps",data=ht,hue="target")
plt.title('resting blood pressure and serum cholestoral')

'''# thalach and oldpeak
plt.subplot(1,3,3)
sns.scatterplot(x="oldpeak", y="thalach",data=ht,hue="target")
plt.title('max heart rate vs ST depression induced by exercise')
'''
plt.show()
```



From the correlation matrix and our initial hypothesis, some features with possible interactions are extracted out for further inspections.

It can be seen that females with heart diseases tend to have higher resting blood pressure compared to those without, whereas males have around the same resting blood pressure distribution with or without heart diseases. There might be an interaction term between *trestbps* and *sex*, suggesting that females are more prone to the effect of high blood pressure compared to males. According to the American Heart Association, women are more likely than men to get high blood pressure at 65 and older, which can lead to higher chance of heart attacks. The relationship between age and sex was also investigated but did not show apparent interaction with respect to heart diseases.

Other obvious interactions lie between *thal*, *chol*, and *trestbps*. As having thalassemia with reversible effect elevates both serum cholesterol and blood pressure, the likelihood of developing heart diseases is higher. In contrast, normal patients have around the same mean cholesterol and blood pressure level with or without heart diseases.

Regarding continuous predictors, it can be seen that there is a general decreasing trend between age and maximum heart rate, as the maximum heart rate naturally decreases when age increases and the cardiopulmonary capacity decreases. There is also a slightly positive correlation between resting blood pressure and serum cholesterol. As cholesterol level increases, there is accumulation of cholesterol plaque and calcium. The arteries are hardened and narrowed, which requires heart to pump the blood harder, resulting in higher blood pressure and therefore higher risk of heart diseases.

6.2.3 Conclusion

In the exploratory data analysis, several risk factors or warning signals related to heart diseases have been identified.

It has been well established that higher blood pressure, cholesterol level, and fasting blood sugar lead to higher chance of heart diseases which are supported by the observations. To mitigate the risks, once could change to a healthy diet, avoiding excessive smoking or drinking, and doing more exercises.

For intrinsic characteristics, men are generally more likely to suffer from heart diseases, but females are more prone to high blood pressure which could lead to heart problems. Regarding family history, the inherited blood disorder thalassemia is also a vital influencer as it decreases the ability of blood transferring oxygen. Out of three different types of thalassemia, the reversible defect is the riskiest with most patients exhibiting higher cholesterol level and blood pressure, resulting in higher risk of heart diseases.

One should be alerted when experiencing chest pains, especially for typical angina. Within the resting ECG results, one should particularly watch out for left ventricular hypertrophy or ST-T wave abnormality. Having a much higher number of ST depression induced by exercises is also a bad indication that heart may not be normal functioning under exercise. In addition, both downsloping and flat slope of peak exercise ST segment are bad signs with higher occurrence of heart diseases. It is therefore advised to conduct regular checks for early detection of possible heart conditions.

The risk of having heart diseases will be reduced with a high maximum heart rate. Doing exercise on a regular basis is a great way to help increase the maximum heart rate and aerobic capacity. It is also advised from the doctors to set a percentage of individual maximum heart rate as a target during exercise. On the other hand, one should be alerted if experiencing angina during or after exercise which could be indication of heart diseases.

6.2.4 Further Studies

It should be noted that exploratory plots do not suggest definitive relationships but provide possible predictive features and interaction terms with respect to the outcome. The significance of each predictive feature needs to be further verified with chi-square tests. Also, there exist some outliers with extreme high or low values in different features that requires further investigated to be applied in the model building process.

To quantify individual effects, a logistic regression could be applied. Other classification techniques including k-nearest neighbor, support vector machines, random forest, decision tree algorithm could be used and compared for better predictivity. Although there is a trade-off between model's interpretability and flexibility, it is aimed in the future studies we can explain the possible mechanisms of heart diseases occurrence with a still robust predictive model for early detections.

6.2.5 References

1. Heart Disease UCI. (2020). Retrieved 19 January 2020, from [https://www.kaggle.com/ronitf/heart-disease-uci]
2. What Causes Heart Disease? Explaining the Model | Kaggle. (2020). Retrieved 20 January 2020, from [https://www.kaggle.com/tentotheminus9/what-causes-heart-disease-explaining-the-model]
3. Analyzing the Heart Disease | Kaggle. (2020). Retrieved 20 January 2020, from [https://www.kaggle.com/kralmachine/analyzing-the-heart-disease/notebook]
4. Thalassemia - Symptoms and causes. (2020). Retrieved 19 January 2020, from [https://www.mayoclinic.org/diseases-conditions/thalassemia/symptoms-causes/syc-20354995]

5. G, W. (2020). Why do men get more heart disease than women? An international perspective. - PubMed - NCBI. Retrieved 20 January 2020, from [https://www.ncbi.nlm.nih.gov/pubmed/10863872]
6. ST depression. (2020). Retrieved 20 January 2020, from [https://en.wikipedia.org/wiki/ST_depression]
7. Publishing, H. (2020). Throughout life, heart attacks are twice as common in men than women - Harvard Health. Retrieved 20 January 2020, from [https://www.health.harvard.edu/heart-health/throughout-life-heart-attacks-are-twice-as-common-in-men-than-women]
8. Angina pectoris: how has the clinical presentation evolved? Is it still the same. (2020). Retrieved 20 January 2020, from [https://www.escardio.org/Journals/E-Journal-of-Cardiology-Practice/Volume-15/Angina-pectoris-how-has-the-clinical-presentation-evolved-Is-it-still-the-same-today-as-it-was-several-years-ago]
9. Publishing, H. (2020). What your heart rate is telling you - Harvard Health. Retrieved 20 January 2020, from [https://www.health.harvard.edu/heart-health/what-your-heart-rate-is-telling-you]
10. High Blood Pressure and Women. (2020). Retrieved 20 January 2020, from [https://www.heart.org/en/health-topics/high-blood-pressure/why-high-blood-pressure-is-a-silent-killer/high-blood-pressure-and-women]
11. Diseases Caused By High Cholesterol | Cleveland Clinic. (2020). Retrieved 20 January 2020, from [https://my.clevelandclinic.org/health/articles/11918-cholesterol-high-cholesterol-diseases]

[ ]: