

## 빅데이터 기반 AI 응용 솔루션 개발자 전문과정

### 교과목명 : 선형대수학

- 평가일 : 22.11.04
- 성명 : 신창훈
- 점수 : 100

Q1. 다음을 수행하세요.

1. np.arange(9)로 구성된 정방행렬
2. 주대각선 원소가 [1,2,3]인 대각 정방행렬
3. 3행 3열로 구성된 항등행렬(단위행렬)
4. 3행 1열로 구성된 영벡터

```
# 1). np.arange(9)로 구성된 정방행렬
import numpy as np
```

```
arr = np.arange(9).reshape(3,3)
arr
```

```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

```
# 2) 주대각선 원소가 [1,2,3]인 대각 정방행렬
np.diag([1,2,3])
```

```
array([[1, 0, 0],
       [0, 2, 0],
       [0, 0, 3]])
```

```
# 3) 3행 3열로 구성된 항등행렬(단위행렬)
np.eye(3,3)
```

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

```
# 4) 3행 1열로 구성된 영벡터
```

```
np.zeros([3,1])
```

```
array([[0.],
       [0.],
       [0.]])
```

Q2. mt1 행렬에 대하여 다음을 수행하세요.

1. 역행렬 mt2를 구하고
2. mt1과 mt2를 내적하여 단위행렬을 출력

```
np.random.seed(0)
mt1 = np.random.randint(1,4,size = (3,3))
mt1
```

```
array([[1, 2, 1],
       [2, 2, 3],
       [1, 3, 1]])
```

```
mt2=np.linalg.inv(mt1)
mt2
```

```
array([[ 7., -1., -4.],
       [-1.,  0.,  1.],
       [-4.,  1.,  2.]])
```

```
mt1.dot(mt2)
```

```
array([[1., 0., 0.],
       [0., 1., 0.],
       [0., 0., 1.]])
```

Q3. 행렬식의 기하학적 성질은 부피를 의미하며  $2 \times 2$  행렬 A의 행렬식은 일종의 넓이로 이해할 수 있다.  $\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$  배열에 대하여 행렬식을 이용하여 면적을 구하세요.

```
arr = [[2,0],[0,2]]
np.linalg.det(arr)
```

```
4.0
```

Q4.  $f(x) = x^2 - 4x$  방정식에 대하여 점(3,-3)에서 접선의 기울기를 구하고 그래프로 표현하면 아래와 같다. scipy.misc 패키지를 이용하여 그 점에서의 기울기를 구하세요.

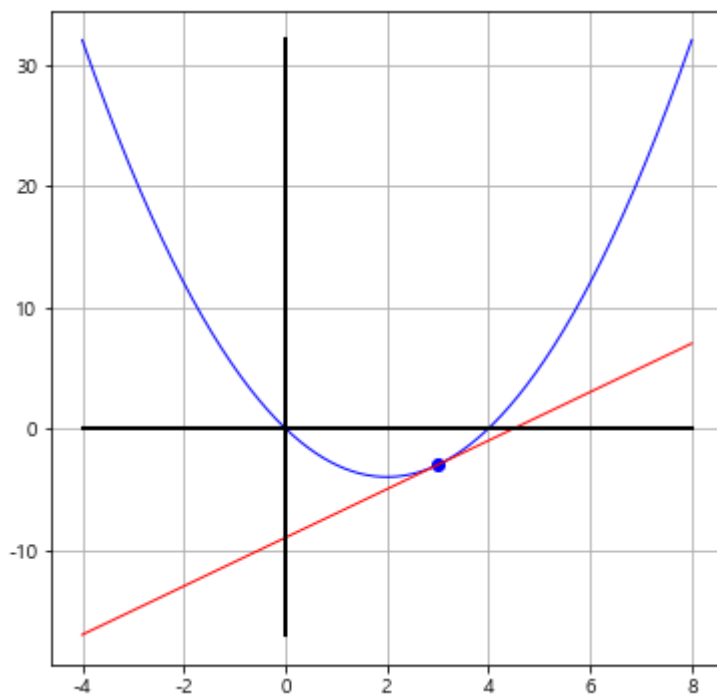
```
import matplotlib.pyplot as plt
```

```
import numpy as np
import scipy as sp
import pandas as pd
import statsmodels.api as sm
import sklearn as sk
import matplotlib as mpl
```

```
mpl.rc('font', family='Malgun Gothic') #한글 폰트 설정
mpl.rcParams['axes.unicode_minus'] = False # 마이너스 단위 처리
```

```
# 접선의 방정식 : 미분의 정의에 따라  $y = f(x)$ 라는 그래상의 점  $(a, f(a))$ 에서 접선이 기울기는  $f'(a)$ 
def f(x):
    return(x**2 - 4*x)
def g(x):
    return(2*x -9)

x = np.linspace(-4, 8, 500)
fig = plt.figure(figsize=(6,6))
plt.scatter([3],[-3],c='b')
plt.plot(x, f(x), '-', lw=1, c='b')
plt.plot(x, g(x), '-', lw=1, c='r')
plt.plot([x.min(), x.max()], [0, 0], lw=2, c='k')
plt.plot([0, 0], [g(x).min(), f(x).max()], lw=2, c='k')
plt.grid()
```



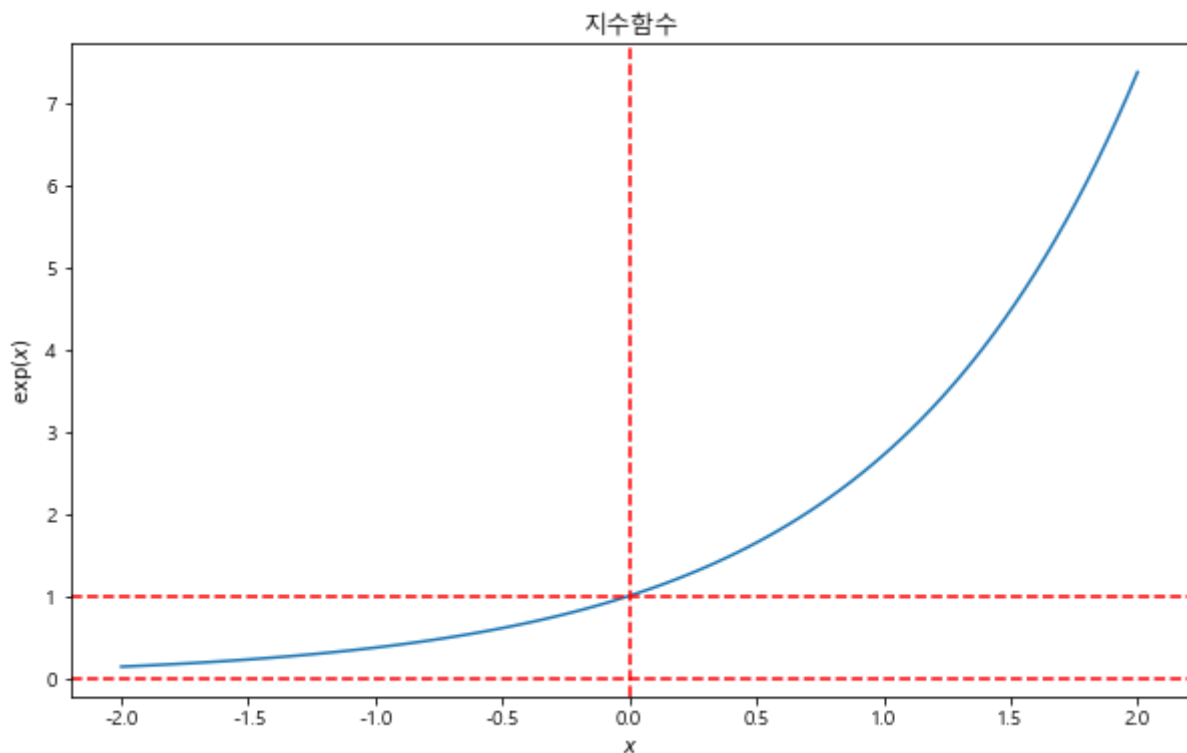
```
from scipy.misc import derivative
def f(x):
    return(x**2 - 4*x)
derivative(f, 3, dx=-3)

2.0
```

Q5. 아래 지수함수에 대응하는 로그함수를 그래프로 표현하세요.

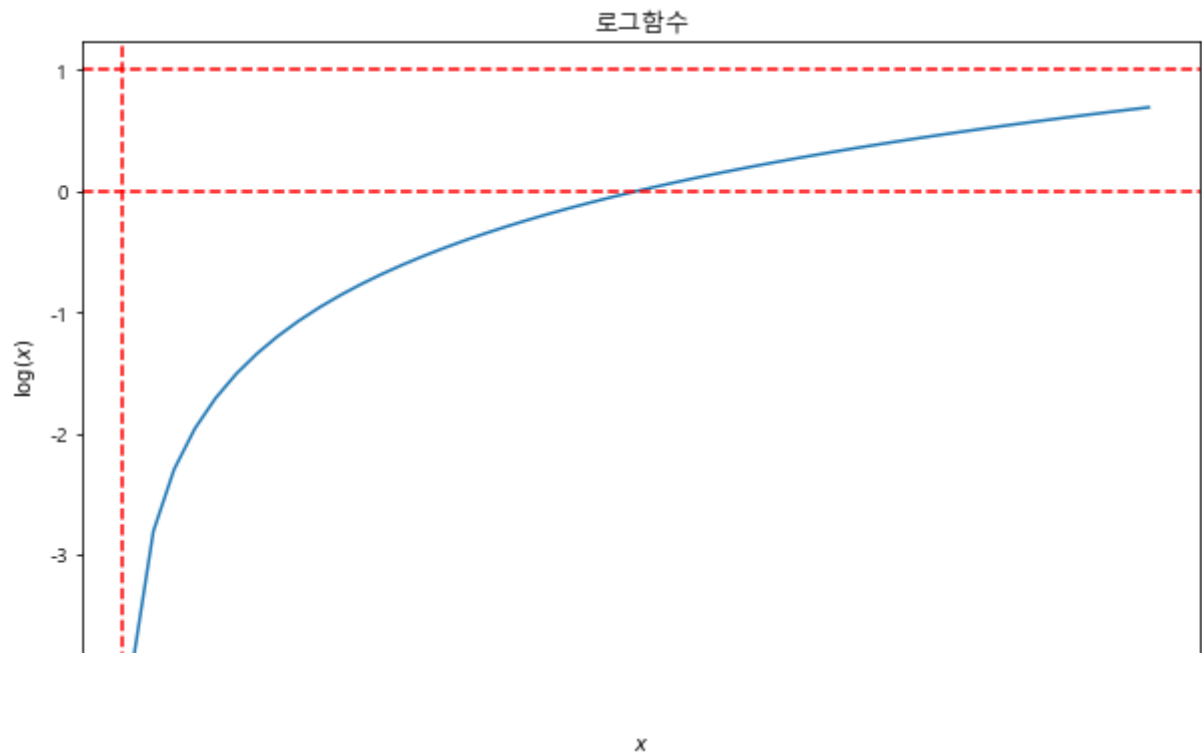
```
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
mpl.rc('font', family='Malgun Gothic') #한글 폰트 설정
mpl.rcParams['axes.unicode_minus'] = False # 마이너스 단위 처리
```

```
# 지수함수의 그래프 1
xx = np.linspace(-2, 2, 100)
yy = np.exp(xx)
plt.figure(figsize=(10,6))
plt.title("지수함수")
plt.plot(xx, yy)
plt.axhline(1, c='r', ls="--")
plt.axhline(0, c='r', ls="--")
plt.axvline(0, c='r', ls="--")
plt.xlabel("$x$")
plt.ylabel("$\exp(x)$")
plt.show()
```



```
import warnings
warnings.filterwarnings('ignore')
```

```
# 로그함수의 그래프
xx = np.linspace(-2, 2, 100)
yy = np.log(xx)
plt.figure(figsize=(10,6))
plt.title("로그함수")
plt.plot(xx, yy)
plt.axhline(1, c='r', ls="--")
plt.axhline(0, c='r', ls="--")
plt.axvline(0, c='r', ls="--")
plt.xlabel("$x$")
plt.ylabel("$\log(x)$")
plt.show()
```



Colab 유료 제품 - [여기에서 계약 취소](#)

