Multivariate Analysis Assignment 1 Solutions

January 20, 2021

0.0.1 Multivariate Analysis Assigmment 1 Solutions

- Sayar Ghosh Roy
- Roll Number: 20171047

```
In [1]: import numpy as np
        np.set_printoptions(suppress = True)
```

0.1 *Problem* 1

```
In [2]: y_1 = [35, 35, 40, 10, 6, 20, 35, 35, 35, 30]
        y_2 = [3.5, 4.9, 30.0, 2.8, 2.7, 2.8, 4.6, 10.9, 8.0, 1.6]
        y_3 = [2.80, 2.70, 4.38, 3.21, 2.73, 2.81, 2.88, 2.90, 3.28, 3.20]
```

```
In [3]: y_1 = np.asarray(y_1)
        y_2 = np.asarray(y_2)
        y_3 = np.asarray(y_3)

        Y = np.stack((y_1, y_2, y_3)).T
        print('Data Matrix Dimensions: ' + str(Y.shape))
```

```
Data Matrix Dimensions: (10, 3)
```

```
In [4]: mean_vector = np.mean(Y, axis = 0)
        covariance = np.cov(Y.T)
        correlation = np.corrcoef(Y.T)
```

```
In [5]: print('Mean Vector: ')
        print(np.ndarray.tolist(np.around(mean_vector, 2)))
        print()

        print('Sample covariance matrix:')
        print(np.around(covariance, 2))
        print()

        print('Sample correlation matrix')
        print(np.around(correlation, 2))
```

```
Mean Vector:
[28.1, 7.18, 3.09]

Sample covariance matrix:
[[140.54  49.68   1.94]
 [ 49.68  72.25   3.68]
 [  1.94   3.68   0.25]]

Sample correlation matrix
[[1.   0.49 0.33]
 [0.49 1.   0.86]
 [0.33 0.86 1.  ]]
```

0.2 *Problem* 2

```
In [6]: data = [[51, 36, 50, 35, 42],
                 [27, 20, 26, 17, 27],
                 [37, 22, 41, 37, 30],
                 [42, 36, 32, 34, 27],
                 [27, 18, 33, 14, 29],
                 [43, 32, 43, 35, 40],
                 [41, 22, 36, 25, 38],
                 [38, 21, 31, 20, 16],
                 [36, 23, 27, 25, 28],
                 [26, 31, 31, 32, 36],
                 [29, 20, 25, 26, 25]]

        weights = [[1, 1, 1, 1, 1],
                   [2, -3, 1, -2, -1],
                   [-1, -2, 1, -2, 3]]

        data = np.asmatrix(data)
        weights = np.asmatrix(weights)

        y = {}
        # Stores feature specific data, namely y_1, y_2, ... , y_5
        for index in range(1, 6):
          y[index] = np.asarray(data[:, index - 1].T).reshape(-1)

        z = 3 * y[1] - 2 * y[2] + 4 * y[3] - y[4] + y[5]
        w = y[1] + 3 * y[2] - y[3] + y[4] - 2 * y[5]

        print('z: ' + str(np.ndarray.tolist(z)))
        print('w: ' + str(np.ndarray.tolist(w)))
```

```
z: [288, 155, 224, 175, 192, 242, 236, 192, 173, 144, 146]
w: [60, 24, 39, 98, 4, 51, 20, 58, 47, 48, 40]
```

In [7]:
```python
def unbiased_var(vec):
    variance = np.var(vec)
    size = np.shape(vec)[0]
    unbiased = variance * size / (size - 1)
    return unbiased

z_mean = np.mean(z)
z_var = np.var(z)

w_mean = np.mean(w)
w_var = np.var(w)

print('Mean of z: ' + str(np.around(z_mean, 2)))
print('Unbiased Variance of z: ' + str(np.around(unbiased_var(z), 2)))
print('Biased Variance of z: ' + str(np.around(z_var, 2)))
print()

print('Mean of w: ' + str(np.around(w_mean, 2)))
print('Unbiased Variance of z: ' + str(np.around(unbiased_var(w), 2)))
print('Biased Variance of w: ' + str(np.around(w_var, 2)))
```

```
Mean of z: 197.0
Unbiased Variance of z: 2084.0
Biased Variance of z: 1894.55

Mean of w: 44.45
Unbiased Variance of z: 605.67
Biased Variance of w: 550.61
```

In [8]:
```python
s_zw = np.cov(z, w)
r_zw = np.corrcoef(z, w)

print('Covariance between z and w:')
print(np.around(s_zw, 2))
print()

print('Correlation between z and w:')
print(np.around(r_zw, 2))
```

```
Covariance between z and w:
[[2084.     40.2 ]
 [  40.2   605.67]]

Correlation between z and w:
[[1.   0.04]
```

 [0.04 1.]]

```
In [9]: print('Dimension of Matrix \'data\':')
        print(np.shape(data))
        print()

        print('Dimension of Matrix \'weights\':')
        print(np.shape(weights))
        print()

        Z = np.matmul(weights, data.T)
        print('Dimension of Combination Matrix \'Z\':')
        print(np.shape(Z))
        print()

        print('Matrix Z.T = ')
        print(np.around(Z, 2))
        print()

        mean_Z = np.mean(Z.T, axis = 0)
        s_Z = np.cov(Z)
        r_Z = np.corrcoef(Z)

        print('Mean of Z: ')
        print(np.ndarray.tolist(np.around(mean_Z, 2).reshape(-1)))
        print()

        print('Covariance of Z')
        print(np.around(s_Z, 2))
        print()

        print('Correlation of Z')
        print(np.around(r_Z, 2))
```

```
Dimension of Matrix 'data':
(11, 5)

Dimension of Matrix 'weights':
(3, 5)

Dimension of Combination Matrix 'Z':
(3, 11)

Matrix Z.T =
[[ 214  117  167  171  121  193  162  126  139  156  125]
 [ -68  -41  -55  -87  -24  -77  -36  -12  -48 -110  -54]
 [ -17    6  -24  -69   29  -14   15  -41  -21  -13  -21]]
```

```
Mean of Z:
[153.73, -55.64, -15.45]

Covariance of Z
[[ 995.42 -502.09 -211.04]
 [-502.09  811.45  268.08]
 [-211.04  268.08  702.87]]

Correlation of Z
[[ 1.   -0.56 -0.25]
 [-0.56  1.    0.35]
 [-0.25  0.35  1.  ]]
```

0.3 *Problem* 3

In [10]:
```
y_1 = [81, 95, 94, 104, 100, 76, 91, 110, 99, 78, 90, 73, 96, 84, 74,
       98, 110, 85, 83, 93, 95, 74, 95, 97, 72]

y_2 = [80, 97, 105, 90, 90, 86, 100, 85, 97, 97, 91, 87, 78, 91, 86,
       80, 90, 99, 85, 90, 91, 88, 95, 91, 92]

x_1 = [356, 289, 319, 356, 323, 381, 350, 301, 379, 296, 353, 306, 290, 371, 312,
       393, 364, 359, 296, 345, 378, 304, 347, 327, 386]

x_2 = [124, 117, 143, 199, 240, 157, 221, 186, 142, 131, 221, 178, 136, 200, 208,
       202, 152, 185, 116, 123, 136, 134, 184, 192, 279]

x_3 = [55, 76, 105, 108, 143, 165, 119, 105, 98, 94, 53, 66, 142, 93, 68,
       102, 76, 37, 60, 50, 47, 50, 91, 124, 74]
```

In [11]:
```
y_1 = np.asarray(y_1) * (0.01)
y_2 = np.asarray(y_2)
x_1 = np.asarray(x_1)
x_2 = np.asarray(x_2)
x_3 = np.asarray(x_3)
data = np.stack((y_1, y_2, x_1, x_2, x_3))

# Checking Shapes
print('Dimensions: ')
print('y_1: ' + str(y_1.shape))
print('y_2: ' + str(y_2.shape))
print('x_1: ' + str(x_1.shape))
print('x_3: ' + str(x_2.shape))
print('x_3: ' + str(x_3.shape))
print('Data: ' + str(data.shape))
```

```
u = 2 * y_1 - y_2
v = 2 * x_1 - 3 * x_2 + x_3
```

```
Dimensions:
y_1: (25,)
y_2: (25,)
x_1: (25,)
x_3: (25,)
x_3: (25,)
Data: (5, 25)
```

In [12]:
```
y_1
```

Out[12]:
```
array([0.81, 0.95, 0.94, 1.04, 1.  , 0.76, 0.91, 1.1 , 0.99, 0.78, 0.9 ,
       0.73, 0.96, 0.84, 0.74, 0.98, 1.1 , 0.85, 0.83, 0.93, 0.95, 0.74,
       0.95, 0.97, 0.72])
```

In [13]:
```
mean_data = np.mean(data.T, axis = 0)
cov_data = np.cov(data)

print('For the \'data\' Matrix: ')
print()

print('Mean Vector:')
print(np.ndarray.tolist(np.around(mean_data, 3).reshape(-1)))
print()

print('Covariance Matrix')
print(np.around(cov_data, 3))
```

```
For the 'data' Matrix:

Mean Vector:
[0.899, 90.44, 339.24, 172.24, 88.04]

Covariance Matrix
[[   0.013    0.046    0.274   -0.2      1.069]
 [   0.046   42.507    8.973   17.432  -16.81 ]
 [   0.274    8.973 1122.44   512.69   -16.802]
 [  -0.2     17.432  512.69  1853.19   305.032]
 [   1.069  -16.81   -16.802  305.032 1129.457]]
```

In [14]:
```
cov_uv = np.cov(u, v)
cor_uv = np.corrcoef(u, v)

print('Covariance between u and v:')
print(np.around(cov_uv, 2))
```

```
        print()

        print('Correlation between u and v:')
        print(np.around(cor_uv, 2))
```

```
Covariance between u and v:
[[   42.37    55.59]
 [   55.59 14248.25]]

Correlation between u and v:
[[1.   0.07]
 [0.07 1.  ]]
```

```
In [15]: u_1 = y_1 + y_2
         u_2 = y_1 - y_2
         v_1 = x_1 + x_2 + x_3
         v_2 = x_1 - 2 * x_2 + 2 * x_3

         U = np.stack((u_1, u_2))
         V = np.stack((v_1, v_2))

         print('Dimension of')
         print('U: ' + str(U.shape))
         print('V: ' + str(V.shape))
```

```
Dimension of
U: (2, 25)
V: (2, 25)
```

```
In [16]: print('Covariance Matrix for \'u\':')
         print(np.around(np.cov(U), 2))
         print()

         print('Covariance Matrix for \'v\':')
         print(np.around(np.cov(V), 2))
         print()

         print('Covariance Matrix of \'u\' and \'v\':')
         print(np.around(np.cov(U, V), 2))
```

```
Covariance Matrix for 'u':
[[ 42.61 -42.49]
 [-42.49  42.43]]

Covariance Matrix for 'v':
[[5706.93 -888.12]
 [-888.12 8494.81]]
```

```
Covariance Matrix of 'u' and 'v':
[[  42.61  -42.49   10.74  -56.7 ]
 [ -42.49   42.43   -8.45   62.32]
 [  10.74   -8.45 5706.93 -888.12]
 [ -56.7    62.32 -888.12 8494.81]]
```

```
In [17]: # ^_^ Thank You
```

4) Let each y be a p-dimensional vector.

There are n samples of $y \rightarrow y_1, y_2, \ldots, y_n$ (each being a p-dimensional vector).

$\therefore$ We have the following data matrix:

$$Y = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix}, \text{ centered Data Matrix } Y_c = \left(I_n - \frac{1}{n} J_n\right) Y = Y - \frac{1}{n} J_n Y$$

$$= \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix} - \begin{bmatrix} \bar{y}_1 & \bar{y}_2 & \cdots & \bar{y}_p \\ \bar{y}_1 & \bar{y}_2 & \cdots & \bar{y}_p \\ \vdots & \vdots & \ddots & \vdots \\ \bar{y}_1 & \bar{y}_2 & \cdots & \bar{y}_p \end{bmatrix} = \begin{bmatrix} y_{11}-\bar{y}_1 & y_{12}-\bar{y}_2 & \cdots & y_{1p}-\bar{y}_p \\ y_{21}-\bar{y}_1 & y_{22}-\bar{y}_2 & \cdots & y_{2p}-\bar{y}_p \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1}-\bar{y}_1 & y_{n2}-\bar{y}_2 & \cdots & y_{np}-\bar{y}_p \end{bmatrix}$$

Now, by definition of sample covariance matrix:

$$S = \frac{1}{n-1}\left(Y^T Y - \frac{1}{n} Y^T J Y\right) \quad \{Y^T : p \times n\}$$

As: $S_{p\times p} = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})(y_i - \bar{y})^T$

$$= \frac{1}{n-1}\left\{\sum_{i=1}^{n}(y_i y_i^T) + n\bar{y}\bar{y}^T - \sum_{i=1}^{n} y_i \bar{y}^T - \bar{y}\sum_{i=1}^{n} y_i\right\}$$

$$-\sum_{i=1}^{n} y_i y_i^T = Y^T Y, \quad n\bar{y}\bar{y}^T = \frac{Y^T J Y}{n} \quad \left\{\text{As } \bar{y} = Y^T \frac{j}{n}, \ \bar{y}\bar{y}^T = \frac{Y^T j}{n}\frac{j^T Y}{n}\right\}$$

where $j = [1\ 1 \ldots 1]^T$

$$\therefore S = \frac{1}{n-1} \cdot Y^T \left\{I - \frac{1}{n} J\right\} Y$$

Also, $\left\{I - \frac{1}{n}J\right\}$ is an idempotent matrix $\rightarrow$

$$\cdot\left(I - \frac{1}{n}J\right)^T\left(I - \frac{1}{n}J\right) = I^T I - \frac{I^T}{n} J - \frac{J^T}{n} \cdot I + \frac{1}{n^2} J^T J$$

$$= I - \frac{J}{n} - \frac{J}{n} + \frac{1}{n^2} \cdot \cancel{n} \cdot J = I - \frac{1}{n} \cdot J$$

$$\therefore S = \frac{1}{n-1} Y^T \left(I - \frac{1}{n}J\right)^T \left(I - \frac{1}{n}J\right) Y$$

$$= \frac{1}{n-1}\left\{\left(I - \frac{1}{n}J\right)Y\right\}^T \left\{\left(I - \frac{1}{n}J\right)Y\right\} \quad [\text{As } (XY)^T = Y^T X^T]$$

$$= \frac{1}{n-1} Y_c^T Y_c \quad [\text{Proved}]$$

5) Let $\mathbb{Y}_{n\times p} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{bmatrix}$: p-dimensional n datapoints.

$\mathbb{Z} = (z_{ij})_{i=1(1)n,\, j=1(1)p}$ s.t $z_{ij} = (y_{ij} - \bar{y}_j)/s_j$

where $\bar{y}_j = \frac{1}{n}\sum_{i=1}^{n} y_{ij}$, $s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_{ij} - \bar{y}_j)^2 = \frac{1}{n-1} y^T y$

Now, $\mathbb{Z}$'s covariance matrix $= S_Z = \frac{1}{n-1}\mathbb{Z}^T\mathbb{Z} = (S_Z)_{ij}$

$\cdot\ s_{z_j}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(z_{ij} - \bar{z}_j)^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left[\frac{y_{ij} - \bar{y}_j}{s_j} - 0\right]^2$ $\left\{ \text{As } \bar{z}_j = \frac{1}{n}\sum_{i=1}^{n} z_{ij} \right.$

$= \frac{1}{n-1}\cdot\frac{1}{s_j^2}\sum_{i=1}^{n}(y_{ij} - \bar{y}_j)^2$ $\quad = \frac{1}{n s_j}\left(\sum_{i=1}^{n} z_{ij} - \bar{y}_j \cdot n\right)$

$\left. = 0 \right\}$

$\cdot\ s_{z_{jk}} = \frac{1}{n-1}\sum_{i=1}^{n}(z_{ij} - \bar{z}_j)(z_{ik} - \bar{z}_k)$

$= \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{y_{ij} - \bar{y}_j}{s_j}\right)\left(\frac{y_{ik} - \bar{y}_k}{s_k}\right)$ $\quad \{\text{As } \bar{z}_j = \bar{z}_k = 0\}$

$= \frac{1}{n-1}\cdot\frac{1}{s_j s_k}(y_{ij} - \bar{y}_j)(y_{ik} - \bar{y}_k).$

Now, consider $\mathbb{R} : (r)_{ij}$: the correlation coefficient matrix of y's

By definition of $\mathbb{R}$: $r_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}\frac{(y_{ki} - \bar{y}_i)\cdot(y_{kj} - \bar{y}_j)}{s_i \cdot s_j}$

By comparing with $(s_z)_j^2 = (s_z)_{jj}\ \forall j = 1(1)n$

and $(s_z)_{jk}\ \forall j = 1(1)n, k = 1(1)n, j \neq k$

$\rightarrow (s_z)_{ij} = r_{ij}\ \forall i = 1(1)n, j = 1(1)n$

$\Longleftrightarrow S_Z = \mathbb{R}$ [Proved]

6) y: Random Vector

Given : Population Mean of y's $= \mu \Rightarrow E[y] = \mu$

and : Covariance matrix of $y = \Sigma$

- By definition of covariance:

$$\Sigma = E[(y-\mu)^2] = E[(y-\mu)(y-\mu)^T]$$
$$= E[(y-\mu)(y^T-\mu^T)] = E[yy^T - y\mu^T - \mu y^T + \mu\mu^T]$$
$$= E[yy^T] - E[y\mu^T] - E[\mu y^T] + E[\mu\mu^T]$$

{By linearity of expectations}

$$= E[yy^T] - E[y]\mu^T - \mu E[y^T] + \mu\mu^T E[1]$$
$$= E[yy^T] - \mu\mu^T - \mu(E[y])^T + \mu\mu^T \quad \{\text{As } E[y]=\mu\}$$
$$= E[yy^T] - \cancel{\mu\mu^T} - \mu\mu^T + \cancel{\mu\mu^T}$$
$$= E[yy^T] - \mu\mu^T \quad \{\text{Proved}\}$$

7) x is a random vector, a is a constant

Let v be the vector $x-a$. Note : v is also a random vector. Let us denote $E[v]$ by $\bar{v}$.

Now, $\text{cov}(x-a) = \text{cov}(v) = E[(v-\bar{v})(v-\bar{v})^T]$

$$= E[vv^T] - \bar{v}\bar{v}^T \quad \{\text{By Result Proved in } 6\}$$

Now, $\bar{v} = E(v) = E[x-a] = E[x] - a$

And $E[vv^T] = E[(x-a)(x-a)^T] = E[xx^T - xa^T - ax^T + aa^T]$

$$= E[xx^T] - E[x].a^T - a\,E[x^T] + aa^T \quad \text{—— (i)}$$

Also, $\bar{v}\bar{v}^T = (E[x]-a)(E[x]-a)^T$

$$= E[x]\,E[x]^T - E[x]a^T - a\,E[x]^T + aa^T \quad \text{—— (ii)}$$

$\therefore \text{cov}(x-a) = E[vv^T] - \bar{v}\bar{v}^T$

$$= E[xx^T] + \cancel{aa^T} - \{E[x]\,E[x]^T + \cancel{aa^T}\} + E[\cancel{x}]a^T + a\cancel{E[x]}^T$$
$$- E[\cancel{x}]a^T - a\cancel{E[x^T]} \quad \text{— from (i) and (ii)}$$
$$= E[xx^T] - E[x]E[x]^T \quad \{\text{As } E[x]^T = E[x^T]\} = \text{cov}(x) \quad \{\text{Proved}\}$$

{By Result in 6}

Proving the general case: $\text{cov}(ax+by) = a^2\,\text{cov}(x) + b^2\,\text{cov}(y)$
— where a & b are constants.

• $\text{cov}(ax+by) = E[(ax+by)(ax+by)^T] - E[(ax+by)]\,E[(ax+by)]^T$

{by Result proved in 6}

$= E[axx^Ta^T + axy^Tb^T + byx^Ta^T + byy^Tb^T] - (E[ax] + E[by]).$
$(E[x^Ta^T] + E[y^Tb^T])$ {As $(xy)^T = y^Tx^T$}

$= aE[xx^T]a^T + a\,E[xy^T]b^T + b\,E[xy^T]^Ta^T + b\,E[yy^T]b^T$
$- a\,E[x]\,E[x^T]a^T - aE[x]\,E[y^T]b^T - b\,E[y]\,E[x^T]a^T$
$- bE[y]\,E[y^T]\,b^T$

$= a\left(E[xx^T] - E[x]E[x^T]\right)a^T + b\left(E[yy^T] - E[y]E[y^T]\right)b^T$
$+ a\left(E[xy^T] - E[x]\,E[y^T]\right)b^T + b\left(E[xy^T]^T - E[y]\,E[x^T]\right)a^T$

$= a\,\text{cov}(x)\,a^T + b\,\text{cov}(y)\,b^T + a\{E[x].E[y^T] - E[x]\,E[y^T]\}\,b^T$
$+ b\left(E[x^T].E[y] - E[y]E[x^T]\right)a^T$

{if X, Y are independent, $E[XY] = E[X].E[Y]$ by definition}

$= a\;\text{cov}(x)\,a^T + b\,\text{cov}(y)\,b^T.$

Now, if a, b are scalar constants, the above expression further reduces to: $a.a\;\text{cov}(x) + b.b\;\text{cov}(y)$
$= a^2\,\text{cov}(x) + b^2\,\text{cov}(Y).$

i) $\text{cov}(x+Y) = \text{cov}(1.x + 1.y) = 1^2\,\text{cov}(x) + 1^2\,\text{cov}(Y)$
$= \text{cov}(x) + \text{cov}(y)$

ii) $\text{cov}(x-Y) = \text{cov}(1.x + (-1).y) = 1^2\,\text{cov}(x) + (-1)^2\,\text{cov}(y)$
$= \text{cov}(x) + \text{cov}(y)$

8) y_i's $(i = 1(1)n)$ are p-dimensional vectors [Let]

we define $z_i = S^{-1/2}(y_i - \bar{y})$ where $S = \frac{1}{n-1}\sum y_i y_i^T - \frac{n}{n-1}\bar{y}\bar{y}^T$

and $\bar{y}$ is the mean of all y_i(s)

i.e $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

Now, $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}).S^{-1/2}$

$= \frac{S^{-1/2}}{n}\left[\sum_{i=1}^{n} y_i - \sum_{i=1}^{n}\bar{y}\right] = \frac{S^{-1/2}}{n}\sum_{i=1}^{n} y_i - \frac{S^{-1/2}}{n}\sum_{i=1}\bar{y}$

$= S^{-1/2}\bar{y} - \frac{S^{-1/2}}{n}.n\bar{y} = 0$ {Proved}

Sample covariance matrix : $S_z = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \bar{z})(z_i - \bar{z})^T$

$\Rightarrow S_z = \frac{1}{n-1}\sum_{i=1}^{n} z_i z_i^T$ {As $\bar{z} = 0$}

$\Rightarrow S_z = \frac{1}{n-1}\sum_{i=1}^{n} S^{-1/2}(y_i - \bar{y}).(y_i - \bar{y})^T (S^{-1/2})^T$

$= \frac{1}{n-1}S^{-1/2}\left[\sum_{i=1}^{n}\{y_i y_i^T - y_i\bar{y}^T - \bar{y}y_i^T + \bar{y}\bar{y}^T\}\right](S^{-1/2})^T$

$= \frac{S^{-1/2}}{n-1}\left[\sum_{i=1}^{n} y_i y_i^T - \left(\sum_{i=1}^{n} y_i\right)\bar{y}^T - \bar{y}\sum_{i=1}^{n} y_i^T + \sum_{i=1}^{n}\bar{y}\bar{y}^T\right](S^{-1/2})^T$

$= \frac{S^{-1/2}}{n-1}\left[\sum_{i=1}^{n} y_i y_i^T - \cancel{n\bar{y}\bar{y}^T} - \bar{y}n\bar{y}^T + \cancel{n\bar{y}\bar{y}^T}\right](S^{-1/2})^T$

$= \frac{S^{-1/2}}{n-1}\left[\sum_{i=1}^{n} y_i y_i^T - n\bar{y}\bar{y}^T\right](S^{-1/2})^T$

$= S^{-1/2}.\frac{1}{n-1}\left[\sum_{i=1}^{n} y_i y_i^T - n\bar{y}\bar{y}^T\right](S^{-1/2})^T$

$= S^{-1/2}.S.(S^{-1/2})^T = I$ {Proved}

9) $z_i = Ay_i \;\forall\; i = 1(1)n$; y_i(s) and z_i(s) are k-dimensional.

Now, $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$ {By definition}

$= \frac{1}{n}\sum_{i=1}^{n} Ay_i = \frac{1}{n}\cdot A\cdot\sum_{i=1}^{n} y_i = A\cdot\frac{1}{n}\cdot\sum_{i=1}^{n} y_i = A\bar{y}$ {Proved}

Also, $S_z = \frac{1}{n-1}\sum_{i=1}^{n}(z_i - \bar{z})(z_i - \bar{z})^T$

$= \frac{\sum_{i=1}^{n}}{n-1}(Ay_i - A\bar{y})(Ay_i - A\bar{y})^T = \frac{1}{n-1}\cdot\sum_{i=1}^{n} A(y_i - \bar{y})\{A(y_i - \bar{y})\}^T$

$= \frac{1}{n-1}\sum_{i=1}^{n} A(y_i - \bar{y})(y_i - \bar{y})^T A^T$

$= \frac{1}{n-1} A\cdot\left\{\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^T\right\}\cdot A^T$

$= A\cdot\left\{\frac{1}{n-1}\cdot\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^T\right\}A^T$

$= ASA^T$ {Proved, As $S = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})^T$}

10) x: $n\times 1$ vector of random variables.

A: $n\times n$ & symmetric; Given $E[X] = \mu$, $\text{cov}(X) = \Sigma$

Note that: we can factorize $X'AX$ as:

$X'AX = (X-\mu)'A(X-\mu) + \mu'AX + X'A\mu - \mu'A\mu$

Now, $E[X'AX] = E[(X-\mu)'A(X-\mu) + \mu'AX + X'A\mu - \mu'A\mu]$

$= E[(X-\mu)'A(X-\mu)] + E[\mu'AX] + E[X'A\mu] - E[\mu'A\mu]$

$= E[(X-\mu)'A(X-\mu)] + \mu'AE[X] + E[X']A\mu - \mu'A\mu$

$= E[(X-\mu)'A(X-\mu)] + \mu'A\mu + \cancel{\mu'A\mu} - \cancel{\mu'A\mu}$

$= E[(X-\mu)'A(X-\mu)] + \mu'A\mu.$

$= E[(X-\mu)'A(X-\mu)] = E\left[\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}(X_i - \mu_i)(X_j - \mu_j)\right]$

Also; $E[(X-\mu')A(X-\mu)]$ $= \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} E[(X_i - \mu_i)(X_j - \mu_j)]$ {By linearity of expectations}

$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\,\text{cov}(X_i, X_j)$

$= \text{tr}\{A\Sigma\}$ [By definition of trace of a matrix]

$\therefore E[X'AX] = E[(X-\mu)'A(X-\mu)] + \mu'A\mu = \text{tr}\{A\Sigma\} + \mu'A\mu$

[Proved]