1.

a) Anxiety scales: **discrete, ordinal**
b) Circumferences of tree trunks: **discrete, ratio**
c) Temperature as measured in absolute temperature: **continuous, ratio**
d) Color codes from a brush marker color chart: **discrete, nominal** (it can be ordinal if compared two data are same color.)
e) Normal systolic blood pressures: **continuous, ratio**
f) Population counts in counties of North Carolina: **discrete, ratio**
g) US postal codes: **discrete, ordinal**
h) UV (ultra-violet) index scale: **continuous, ratio**
i) An audio volume controlled by up/down buttons on a remote controller: **discrete, ratio**
j) Existence and non-existence of a certain bacteria in a person's blood: **binary, nominal**
k) A percentage of protein in milk: **continuous, ratio**
l) light-years from stars to the earth: **continuous, ratio**
m) The section numbers in a shopping mall: **discrete, ratio**

2.

a)

i. todo, 不太明白为什么要求 normalized 后的均值和方差，难道不是 0 和 1 么

ii. commonly used activation function these days are **sigmoid, tanh, ReLU, Maxout**. To transform a variable to range(-1, 1), we can choose **tanh:**

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

b)

i. one-hot: **discrete, nominal**
word vector: **continuous, nominal** (very unlikely there will be overlap between two word vectors, but it's not imposibble. Just assume they won't overlap)

ii. ***one-hot encoding***
advantage:
- ✧ very easy to represent a word
- ✧ more understandable semantical meaning

disadvantage:
- ✧ extremly sparse and high dimensional vector. Easily causes Curse of Dimensionality.
- ✧ Unable to show similarity among words.

Distributed word vector:
advantage:
- ✧ Normally only hundreds of dimensions, easy to calculate.

- ✧ Can calculate similarity among words (similar words have similar vector)

disadvantage:

- ✧ Need some extra time to train or calculate the word vector.
- ✧ Not easy for human to understand its semantical meaning.

3.

a) **Complete-case analysis** will eliminate some useful data, and **missing value imputation** will bring extra noise.

b) In this case, we need to deal with unbalanced data.

4.

a)

```
> A = diag(3)
> A
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

b)

```
> A[,2] = 3
> A
     [,1] [,2] [,3]
[1,]    1    3    0
[2,]    0    3    0
[3,]    0    3    1
```

c)

```
> sum = 0
> for(i in A) {sum = i + sum}
> sum
[1] 11
```

C %*% Q

d)

```
> sum(A[3,])
[1] 4
> sum(diag(A))
[1] 5
> sum(A[, 2])
[1] 9
```

e)

```
> B = matrix(rnorm(25, mean=7, sd=1), ncol=5, nrow=5)
> B
         [,1]     [,2]     [,3]     [,4]     [,5]
[1,] 8.675211 7.831911 7.449289 6.543275 8.044872
[2,] 7.227597 6.976875 7.025619 7.226987 9.122406
[3,] 6.435951 7.505107 7.994070 7.422234 8.470869
[4,] 7.354461 5.680904 7.114506 8.105177 5.384046
[5,] 7.640132 7.042374 5.903940 5.878043 6.938493
```

f)

```
> C = rbind(B[1,] - B[2,], B[3,] + B[4,])
> C
          [,1]       [,2]       [,3]       [,4]      [,5]
[1,]  1.447614  0.8550359  0.4236691 -0.6837121 -1.077534
[2,] 13.790413 13.1860115 15.1085763 15.5274109 13.854915
```

g)

```
> Q = matrix(diag(2:6), ncol=5)
> C %*% Q
          [,1]      [,2]      [,3]     [,4]      [,5]
[1,]  2.895228  2.565108  1.694677 -3.41856 -6.465203
[2,] 27.580825 39.558034 60.434305 77.63705 83.129490
```

h)

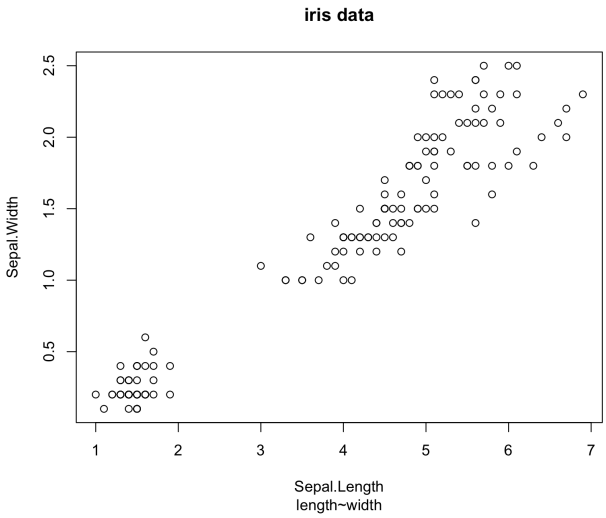
```
> X = c(1, 3, 5, 8)
> Y = c(5, 3, 2, 1)
> covMat = CovMat2D(X, Y)
> covMat
          [,1]      [,2]
[1,]  8.916667 -4.916667
[2,] -4.916667  2.916667
```

i)

```
> mean(X^2) == mean(X)^2 + var(X)
[1] FALSE
```

5.

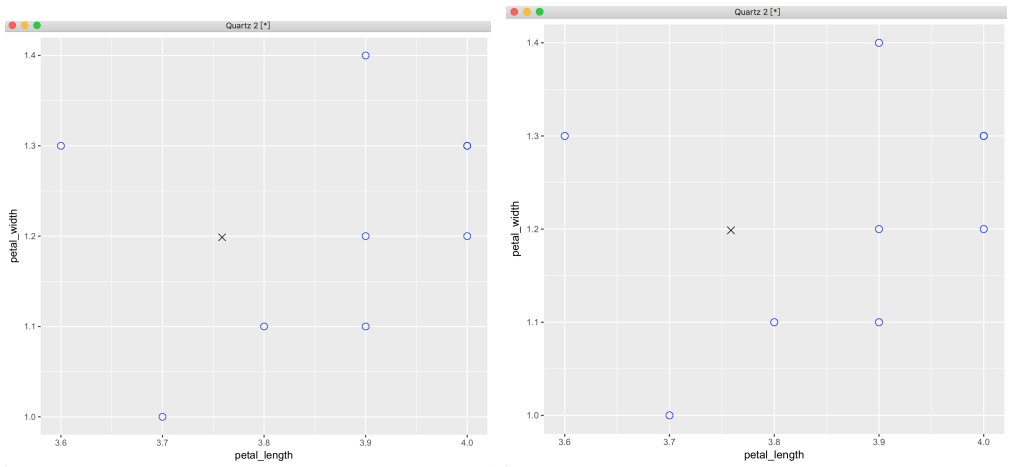
a) data = read.csv(path)
plot(data$petal_length, data$petal_width,type="p", xlab="petal_length", ylab="petal_width")



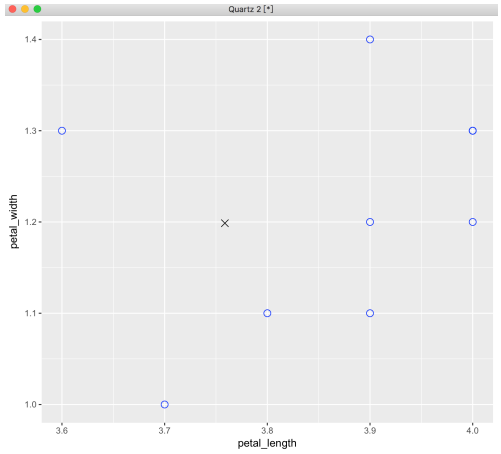
b) mean_length = mean(data$petal_length)
3.758667
mean_width = mean(data$petal_width)
1.198667

c)

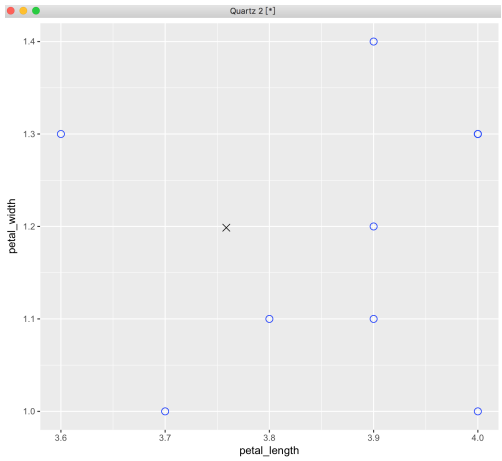
d)



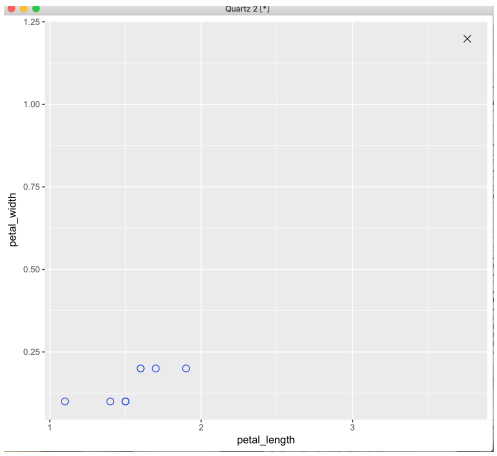
Euclidean Distance



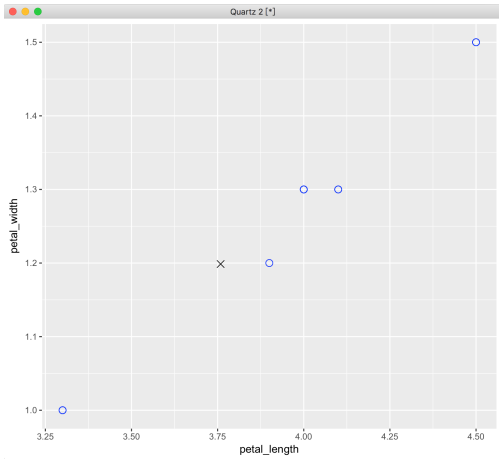
CityBlock Distance



Minkowski Distance(r=3)



Chebyshev Distance



Cosine Distance

Mahalanobis Distance

Look at above figures, we can find that the results of first four figures are very similar, but Cosine Distance and Mahalanobis Distance have big different with them.

For Cosine Distance, if compared two points are A and B, origin porint is O. Then it measures the cosine of $\angle AOB$. SO the nearest points will lie on line OX(X is the point marked with 'X').

6.

a) **mean1** = 82.92707, **mean2** = 90.08649
median1 = 83.645, **median2** = 90.11
standard_deviation1 = 83.9897, **standard_deviation2** = 8.774344
range1 = [70.69, 98.07], **range2** = [81.54, 99.80]

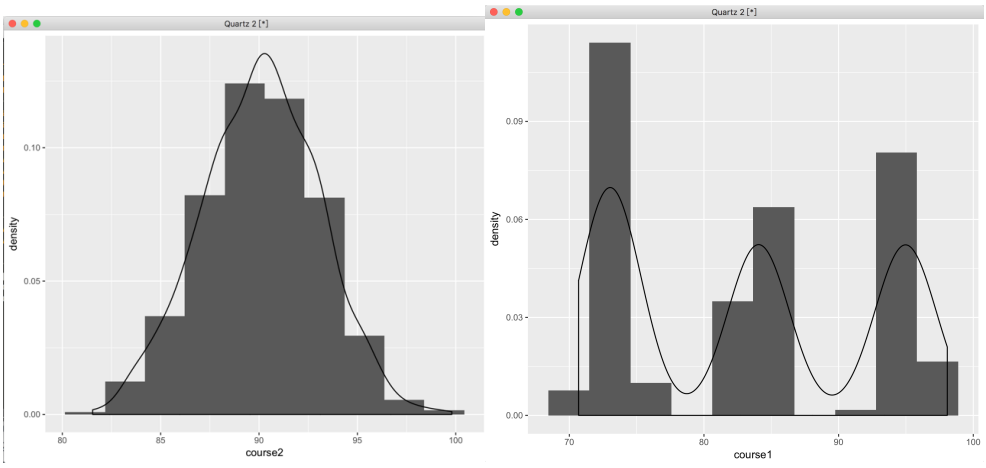
b) course1

25%	50%	75%
73.3475	83.6450	94.0725

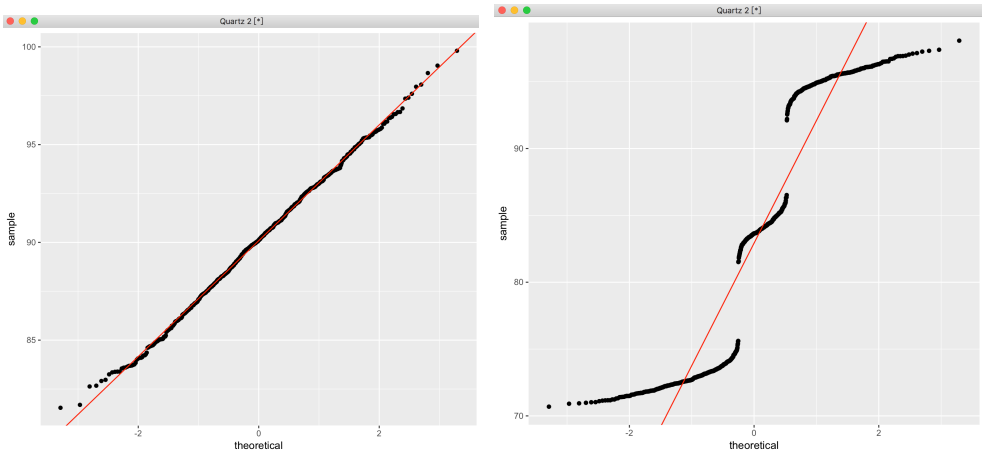
course2

25%	50%	75%
88.1000	90.1100	92.1025

c) ggplot(NULL, aes(x=course2)) +
geom_histogram(bins=10, aes(y=..density..)) +
geom_density()



d) ggplot(NULL) +
stat_qq(aes(sample=course2)) +
geom_abline(intercept=mean2, slope=sqrt(var2), color='red')



QQ plot can be used for examing if