

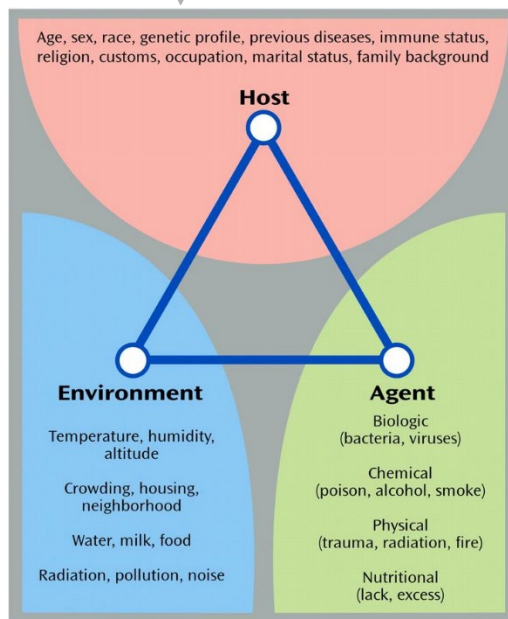
Name: <Zhenyu Zhang>

Chapter 1 – Statistical Epidemiology 1

1 Introduction

Chapter 1 of the guide covers the basics of epidemiology, including key concepts, measures of disease frequency and association, and sources of bias.

2. Epidemiological Triad - The spread of epidemics is believed to be the result of interactions between pathogens, hosts, and the environment; and can be prevented by altering factors that affect exposure and susceptibility. **Case** study and discussion has been done in CHAPTER 1 - PHASE 3 - SOLVE



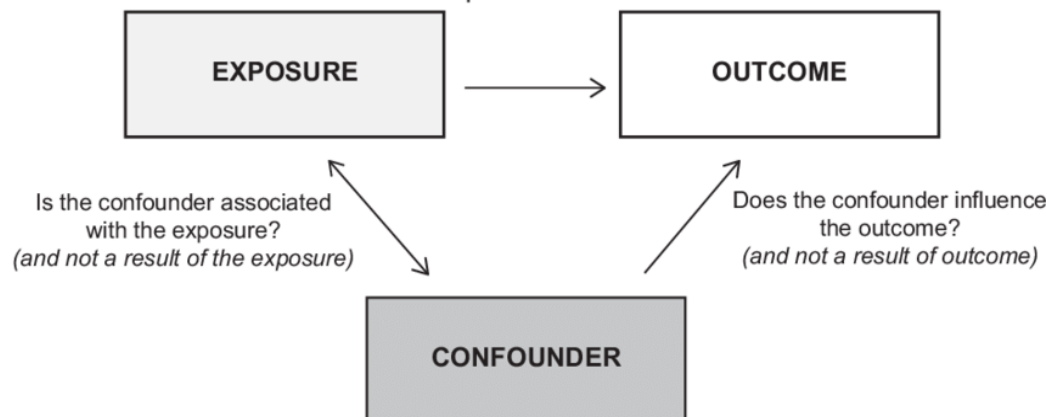
[Diseases and Causation Theories | ESSENCE OF BEING \(wordpress.com\)](#)

3. Confounder must associated with both the outcome and the exposure. A confounder is defined as a concurrent cause of the outcome under investigation that is related to, but not a consequence of, the exposure of interest. Adjusting for it will change the results obtained.

Meeting all three criteria is necessary to consider a variable a confounder:

- The potential confounding factor must be associated with the outcome variable (disease or health status) under investigation.
- The potential confounding variable must be associated with the exposure (treatment) of interest.
- The potential confounding variable must not be on the causal pathway between the exposure and the outcome.

When an extraneous variable meets the criteria for confounding, the analysis of the causal relationship under study needs to be adjusted for this confounding variable to prevent it from biasing the study findings. Should be included even if they are not statistically significant.



[16 Mixed effects: triangular relationship between exposure, outcome,... | Download Scientific Diagram \(researchgate.net\)](#)

4. Randomisation

Randomization is an important aspect of designing and conducting randomized controlled trials (RCTs) to ensure that treatment and control groups are balanced with respect to potential confounders. Successful randomization increases the likelihood that potential confounders are equally distributed over treatment groups, thereby eliminating the association of these factors with the treatment under study and thus their potential for biasing the causal relation under investigation because the first criterion for confounding does not hold anymore. Thus, randomized-controlled studies are sometimes regarded as the gold standard of epidemiologic study designs. R offers various functions to perform randomisation methods. **Part** of the following is completed through additional packages.

1. simple randomisation

The **sample()** function in R can be used to randomly allocate participants to treatment groups. For example, to randomly allocate 50 participants to two groups (treatment and control), we can use the following code to randomly allocate 50 participants to either treatment or control groups.:

```
set.seed(123)
allocation <- sample(c("treatment", "control"), 50, replace=TRUE)
```

2 Block Randomisation

The **blockrand()** function in the **randomizeR** package can be used to perform block randomisation. For example, to allocate participants to treatment and control groups in blocks of size 4, we can use the following code:

```
library(randomizeR)
set.seed(123)
allocation <- blockrand(c(rep("treatment", 2), rep("control", 2)), 50, blocksize = 4)
```

3 Stratified Randomisation

The **stratify()** function in the **randomizeR** package can be used to perform stratified randomisation. For example, to stratify participants by age and allocate them to treatment and control groups, we can use the following code:

```
library(randomizeR)
set.seed(123)
strata <- data.frame(age = c(rep("18-30", 25), rep("31-45", 25)))
allocation <- stratify(strata, c(rep("treatment", 25), rep("control", 25)))
```

4 Minimisation

The **minim()** function in the **randomizeR** package can be used to perform minimisation randomisation. For example, to allocate participants to treatment and control groups based on their age and sex, we can use the following code:

```
library(randomizeR)
set.seed(123)
factors <- data.frame(age = c(25, 35, 40, 45, 50), sex = c("M", "F", "M", "F", "M"))
allocation <- minim(factors, c(rep("treatment", 25), rep("control", 25)))
```

5 Cluster Randomisation

Cluster Randomisation: The **clusterRandomise()** function in the **randomizeR** package can be used to perform cluster randomisation. For example, to allocate hospitals to treatment and control groups, we can use the following code:

```
library(randomizeR)
set.seed(123)
clusters <- data.frame(hospital = c(rep("A", 5), rep("B", 5)))
allocation <- clusterRandomise(clusters, c(rep("treatment", 5), rep("control", 5)))
```

5. Bias

The role of bias in research is to introduce errors or deviations in the results, which can lead to incorrect conclusions or findings. Bias can occur at any stage of the research process, from study design to data collection and analysis.

It is important to evaluate the role of bias in research because it can have a significant impact on the validity and reliability of the study results. It involves assessing the study design, data collection methods, and analysis techniques to identify potential sources of bias. Moreover, statistical methods can be used to quantify the impact of bias on the study results. By understanding and addressing potential sources of bias, researchers can take steps to minimize or eliminate their impact on the study findings.

The types of bias in epidemiology

- Selection bias
- Non-response bias
- (Loss to) Follow up
- Recall / Responder bias
- Interviewer or observer bias
- Surveillance bias
- Random sampling error
 - Incorrectly rejecting the null hypothesis when it is true (a type I or α error).
 - Failing to reject the null hypothesis when it is false (a type II or β error).

Techniques to minimize bias

- Design the study properly.
 - Random sampling
 - Controlling / adjusting for potential confounders
 - Minimizing non-response
 - Use multivariable methods (e.g. modelling).

Blog

This way of learning on Openlearning is new to me. At the beginning, I just found an example of a disease from the Internet, but with the step-by-step guided learning. I benefit a lot from reading and commenting on classmates' answers. Also, I found it was very helpful and efficient for me to apply the concepts of Randomisation and Bias I learned.

I have heard of the epidemiological triad, but it seems like I have always understood it as the triad of infectious disease. This study extended my understanding of epidemiological triad. chronic diseases such as stroke and occupational diseases such as silicosis and other diseases can also fit or partially fit the epidemiological triad model. This model would help to make a more comprehensive analysis and identification of potential risk factors. Therefore, the development of interventions targeting specific factors can more effectively help prevent or reduce the occurrence of diseases or control the further development of diseases.

Understanding Confounding and enumerating Confounder is a very fresh learning process. I have read some literature, few, but it is very interesting to quote a practical case to bring into the analysis. It is also very helpful for me to complete the MCQ part. Same as the two cases in phase 3 helped me practice by using what I learned before to complete the analysis.

I have used a bit of R before; it is not a hard beginning for me at this point. Hoping the coding part can link with the other phases.

Chapter 2 – Statistical Epidemiology 2

1. Causation

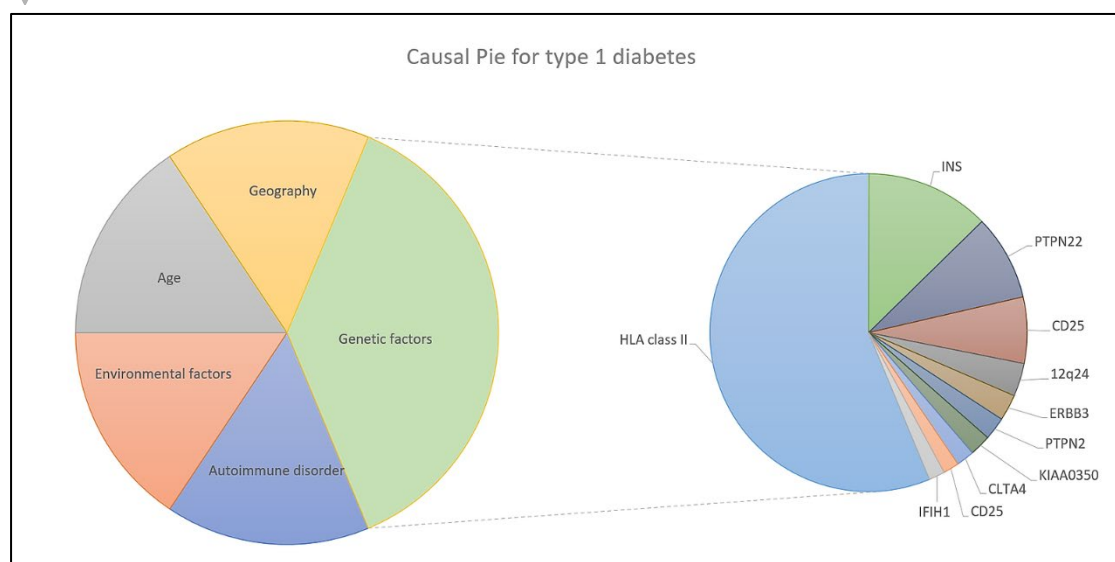
Bradford Hill's criteria indicators a causal link between two associated (or correlated) variables. There are nine factors of Bradford Hill criteria:

1. Strength of association
2. Consistency
3. Specificity
4. Temporality
5. Dose-response relationship (gradient)
6. Plausibility
7. Coherence
8. Experimental evidence
9. Analogy

Rothman's Causal Pie is the model highlights the importance of considering all relevant factors when investigating the causes of a disease.

- Component cause (slice of the pie) - Any one of the set of conditions which are necessary for the completion of a sufficient cause.
- Sufficient causes (the whole pie) – a minimum set of component causes that make the outcome inevitable. A disease may have several sufficient causes (several pies can produce the same disease).
- Necessary cause (found in all cases) - A component cause that is a member of every sufficient cause.

Rothman's Causal Pie for Type 1 diabetes:



2. Measurement Errors

Random error

- Type I
- Type II

Systematic error

- Selection Bias
- Measurement bias or Information bias
 - Subject error
 - Instrument error
 - Observer error
- Confounding (https://sph.unc.edu/wp-content/uploads/sites/112/2015/07/nciph_ERIC11.pdf)

Non-differential misclassification occurs when the misclassification of a variable is unrelated to the exposure or outcome being studied. This means that the error in measurement is random and affects both the exposed and unexposed or the diseased and non-diseased groups equally. This can lead to an underestimation or dilution of the true effect of the exposure or outcome.

Differential misclassification, on the other hand, occurs when the misclassification of a variable is related to the exposure or outcome being studied. This means that the error in measurement is not random and affects one group more than the other. This can lead to an overestimation or inflation of the true effect of the exposure or outcome.

Measurement errors can have a significant impact on the results of a study and can affect the validity and reliability of the study's findings. In some cases, measurement errors can be random and unpredictable, which can lead to an underestimation or overestimation of the true association between variables, but they may not necessarily result in bias. However, when the measurement errors are systematic and consistently affect the measurements in a particular direction, they can introduce bias and lead to incorrect conclusions. To minimize the impact of measurement errors on study results, researchers must take appropriate steps to ensure that the measurement methods used are valid and reliable, and that the measurements are accurate and consistent. This may involve using standardized measurement tools, calibrating equipment, conducting training and certification of data collectors, and implementing quality control procedures.

Possible measurement error in scenario of investigating type 1 diabetes:

- Subject error – Examination of dietary exposures and type 1 diabetes essentially based on participants' recall, the introduction timing and duration of breastfeeding, cows' milk-based formula and any type of solid foods and/or cereals (gluten and non-gluten containing) might be recalled in error due to parental anxiety or guilt.
- Instrument error – Blood glucose levels are a key indicator of diabetes. Errors in the measurement of blood glucose levels can lead to incorrect diagnoses directly. The accuracy of the glucose meter, the incorrect calibration of the glucose meter will cause instrument error.
- Observer error – for example, the person measuring the blood glucose is not experienced, does not use the instrument properly, and makes an error in interpreting or recording the blood glucose measurement.

Inaccurate measurement of blood glucose levels can lead to misclassification of individuals with or without diabetes, which directly affecting the estimated the prevalence. In addition, it will affect the accuracy of estimates of associations strength between risk factors and type 1 diabetes, and even whether the associations are positive or negative. That might be the reason of contradictory results produced in many different studies of dietary factors in childhood on the risk for islet autoimmunity and type 1 diabetes.

Possible ways to limit the amount of measurement error that occurs:

- Using high-quality and regularly calibrated glucose meters.
- Standardizing the method of blood sampling and implement complete training for observer.
- Maintaining consistent testing conditions, for example at the same time of day, after a similar meal or fasting period, and at a similar level of physical activity.
- Using multiple measurements to reduce the random measurement variability.

3. Agreement

Bland-Altman plots are commonly used to assess the agreement between two quantitative measures of the same variable. The following steps can be taken to construct a Bland-Altman plot:

1. Collect the measurements: Collect measurements from both methods for each individual in your sample. Make sure the same units of measurement are used for both methods.
2. Calculate the mean and difference: Calculate the mean of the two measurements and the difference between the two measurements for each individual in your sample.
3. Create a scatter plot: Create a scatter plot with the mean on the x-axis and the difference on the y-axis. Each point on the plot represents one individual.
4. Add horizontal lines: Add horizontal lines at the mean difference and at the limits of agreement, which are typically set at ± 2 standard deviations of the differences.
5. Interpret the plot: Look for any patterns in the data that may indicate systematic bias or heteroscedasticity, such as a trend in the differences or increasing variance as the mean increases. Also, check if the majority of the points fall within the limits of agreement.
6. Calculate the correlation coefficient: Calculate the correlation coefficient between the two methods using a correlation test, such as a Pearson correlation or a Spearman rank correlation.

```
clinician_1 <- c(0.3, 0.6, 1.8, 1.2, 0.7, 1.3, 0.7, 0.4, 0.9, 0.1, 1.4, 0.8, 0.2)
clinician_2 <- c(0.6, 1.0, 2.4, 1.6, 1.0, 1.8, 1.0, 0.7, 1.3, 0.3, 1.9, 1.1, 0.4)
# Step 1. Given the measurements into the R vectors clinician_1 and clinician_2

clinician_diff = clinician_2 - clinician_1
data <- data.frame(clinician_1,clinician_2)
clinician_avg <- rowMeans(data, na.rm = TRUE)
clinician_mean_diff <- mean(clinician_diff)
# Step 2 Calculate the mean and difference

plot (clinician_1, clinician_2, type="p")
```

Step 3. This code will create a scatter plot of two sets of data, `clinician_1` and `clinician_2`, using the `plot` function. The `type` argument is set to "p" which means the plot will show points. The plot will allow us to visually inspect the relationship between the two variables.

```
abline(h = clinician_mean_diff, lty=3, col="green")
```

```
clinician_sd_diff <- sd(clinician_diff)
clinician_upper_la <- clinician_mean_diff + 2 * clinician_sd_diff
clinician_lower_la <- clinician_mean_diff - 2 * clinician_sd_diff
```

```
abline(h=clinician_sd_diff, lty=3, col="red")
abline(h=clinician_upper_la, lty=3, col="blue")
abline(h=clinician_lower_la, lty=3, col="blue")
```

Step 4 Add horizontal lines to the plot representing the mean difference (green), the upper and lower limit of agreement (blue) respectively.

```
cor.test(clinician_1, clinician_2)
```

Step 6 Performing a Pearson correlation test between `clinician_1` and `clinician_2` using the `cor.test()` function. The Pearson correlation coefficient measures the linear association between two variables. The test will output the correlation coefficient, p-value, and confidence interval. This test will help us determine the strength and direction of the association between the two variables.

Note: correlation (more on correlation in later chapter) is a measure of association, not a measure of agreement. Furthermore, two measurements can be highly correlated, yet not be in agreement.

4. Screening test.

- ❖ Primary - Prevention of future occurrence in unaffected individuals by removing a cause.
- ❖ Secondary - Prevention of clinical disease by screening, early detection and/or treatment.
- ❖ Tertiary - Prevention of disease by treating cases (disease management).

Gold standard refers to the best available method or test for diagnosing a particular condition. It is considered to be the most accurate and reliable test that can be used as a reference standard against which other diagnostic tests can be compared.

Sensitivity, specificity, positive predictive value (PPV), and negative predictive value (NPV) are measures of diagnostic test accuracy.

Sensitivity is the proportion of true positive cases among all the individuals who actually have the disease or condition. It is calculated as: $\text{Sensitivity} = a / (a + c)$

Specificity is the proportion of true negative cases among all the individuals who do not have the disease or condition. It is calculated as: $\text{Specificity} = b / (b + d)$

PPV is the proportion of true positive cases among all the individuals who test positive for the disease or condition. It is calculated as: $\text{PPV} = a / (a + b)$

NPV is the proportion of true negative cases among all the individuals who test negative for the disease or condition. It is calculated as: $\text{NPV} = d / (d + c)$

Accuracy is another metric used to describe the performance of a screening test, which measures the proportion of true positive and true negative results among all the test results. It is calculated as: $(a + d) / (a + b + c + d)$.

- While accuracy is an important measure, it should be interpreted with caution, especially when the prevalence of the condition being screened for is low. In such cases, a high level of accuracy may be achieved simply because the test is correctly identifying people who do not have the condition (i.e., true negatives), rather than identifying those who do have the condition (i.e., true positives). This can lead to a high number of false negative results and can result in missed diagnoses. Therefore, sensitivity and specificity are typically considered more informative measures of a screening test's performance.

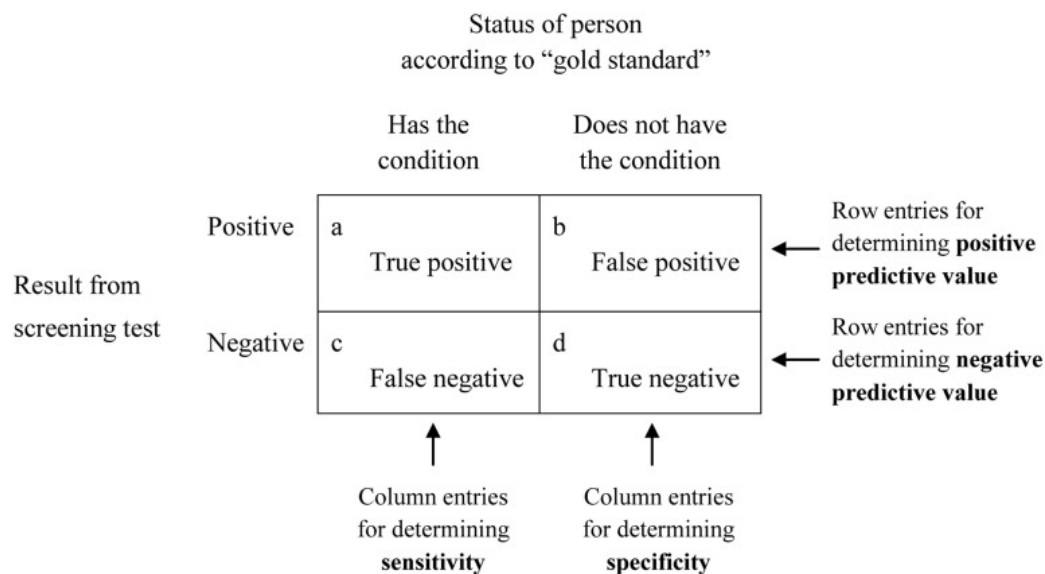


Diagram demonstrating the basis for deriving sensitivity, specificity, and positive and negative predictive values. (Jovel, Patterson et al. 2016) <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5701930/>

These metrics are often cited as percentages or decimal fractions and are used to evaluate the accuracy of a screening test. However, deficiencies in either the reference standard or the screening test can exist, leading to inaccuracies in the results. There can be tradeoffs between sensitivity and specificity, as well as between positive and negative predictive values. These metrics are not fixed attributes of a screening test and can vary depending on the stringency of the test and the prevalence of the target condition in the sample being analysed. Therefore, it is important to thoroughly evaluate the validity of the reference standard and screening test, consider the tradeoffs between metrics, and provide information about all metrics and the sample on which they are based to accurately characterise a screening test.

Please see the Scenario with data (my MCQ) from C9 of why we need confirmation test in the screening

Blog

Those definitions in epidemiology are difficult for me. The mere textual explanation of the concepts and definitions is still not enough for me to fully understand. Additional reading and MCQs are good supplements for me. However, the chapter 3.2 of "Measurement Errors in Epidemiology" is difficult to understand. I do not have enough time to learn this part, so have to put it aside.

Hills Criteria and Rothman's Causal Pie are very helpful to understand and analyse the causal link between two associated variables. Most of the current medical research hotspots are cancer and chronic diseases, which are intertwined by various reasons. This makes them more like summary tools.

Agreement is very interesting, $\text{plot}(x,y)$ is a very useful and powerful tool, it would be better if there are explains the results of Pearson's product-moment correlation.

Screening might be the most important tool for prevention. However, it never occurred to me that screening has sufficient statistics to comparative detect its performance. That sometimes, screening could have a bad PPV leading to unnecessary follow-up testing or bad NPV causing a missed diagnosis. Some certain screening tests still could have utility because they are inexpensive, simple, and have no or low negative side effects.

Chapter 3 – Statistical Epidemiology 3

Incidence

Number at risk is a count of the total people that are feasibly at risk of experiencing the outcome of interest.

$$\text{Incidence} = \frac{\text{Number of new cases in period}}{\text{Number at risk new cases in period}}$$

The number at risk can vary over time, due to people:

- being born or dying.
- being recruited into, or leaving, a study at different times.
- actually getting the disease being studied.

Therefore, we consider ‘person-time’ at risk.

$$\text{Incidence} = \frac{\text{Number of new cases}}{\text{Total person – time at risk}}$$

Prevalence is the number of people who have a disease/outcome amongst a population at a specific time. It is also known as the *point prevalence*, since it is a measure at a point in time.

$$\text{Prevalence} = \frac{\text{Number with disease at certain time}}{\text{Number in population at that certain time}}$$

Standardisation

Standardisation is a statistical method used to adjust crude rates for differences in the population structure (e.g. age or sex) between different groups or populations being compared. The purpose of standardisation is to make the comparison of rates more valid by removing the effect of differences in the population structure. This is achieved by calculating stratum-specific rates (rates for each subgroup defined by age or other factors), and then combining these rates using a weighted average to obtain an overall rate that reflects the age or other characteristic distribution of the reference population.

There are two main types:

- ❖ Direct standardisation, the stratum-specific rates of the study population are applied to the age or other characteristic distribution of the reference population.
- ❖ Indirect standardisation, the stratum-specific rates of the reference population are applied to the age or other characteristic distribution of the study population.

The standardised mortality ratio (SMR) compares the number of the observed deaths against the number of expected deaths.

For indirect standardisation of SMR, a:

- ❖ $\text{SMR} < 100$ means that the mortality (death) rate (MR) is less than ($<$) that of the standard population.
- ❖ $\text{SMR} = 100$ means that the MR is the same as that of the standard population.

❖ SMR > 100 means that the MR is greater than (>) that of the standard population.
expected number of deaths (MR):

$$MR \text{ for the reference population} = \frac{\text{Number of observed deaths in the reference population}}{\text{Population size}}$$

Expected number of deaths = MR × Study population size

$$SMR = \frac{\text{Observed}}{\text{Expected}} \times 100\%$$

The case of IHD data in Hobart:

	Hobart		Australia	
Age group	Observed deaths	Population	Observed deaths	Population
35-44	75	21,100	8,596	3,314,676
45-54	259	16,500	38,600	3,042,810
55-64	742	15,400	103,843	2,894,186
65-74	1089	11,100	175,433	2,208,684

```
IHD = (2165/64100)*1000
```

```
cat("The crude mortality rate of Ischemic heart disease (IHD) of males in Hobart is", IHD, "deaths per 1000")
```

```
Observed_deaths = 75 + 259 + 742 + 1089
```

```
Expected_deaths = 54.72 + 209.31 + 552.55 + 881.66
```

```
SMR = Observed_deaths / Expected_deaths * 100
```

```
cat("\nObserved number of deaths is", Observed_deaths)
```

```
cat("\nExpected number of deaths", Expected_deaths)
```

```
cat("\nThe (indirect) SMR of IHD of males in Hobart is ", SMR, "%", sep = "")
```

The crude mortality rate of Ischemic heart disease (IHD) of males in Hobart is 33.77535 deaths per 1000

Observed number of deaths is 2165

Expected number of deaths 1698.24

The (indirect) SMR of IHD of males in Hobart is 127.4849%

We can conclude that Hobart has a higher rate of male IHD mortality than expected in the Australian population.

Further, we can say that the male IHD mortality in Hobart is about 127% of that expected in the Australian population.

```
Observed deaths in Hobart = 2,165
```

```
Total male population of Hobart = 64,100
```

```
Crude MR Hobart = 2,165 / 64,100 = 33.78 per 1000
```

```
Sum of Australian population x Hobart MR for all strata = 415,681.87
```

```
Total Australian male population = 11,460,356
```

```
SMR = 415,681.87 / 11,460,356 = 36.27 per 1000
```

The Australian male IHD MR = $(326,472 / 11,460,356) = 28.5$ per 1000
 $36.3/28.5 = 127.37\%$

We can conclude that age-standardised, Hobart has a male IHD MR of 36.3 per 1000.

This is higher than the Australia-wide rate of 28.5 per 1000.

Further, we can say that male IHD mortality in Hobart is about 127% of that expected in the Australian population.

Rates, Risks and Odds

Risk

$$\text{Risk} = \frac{\# \text{favourable}}{\text{Total}}$$

- ❖ Same as probability to a statistician
- ❖ Calculated same way as probability

Risk ratio (RR)

also known as the relative risk, is a measure of the association between exposure to a risk factor and an outcome. In this example, the risk ratio compares the risk of stroke, MI, or death in patients who received a high dose of acetylsalicylic acid to those who received a low dose.

$$RR = \frac{\text{Risk}_A}{\text{Risk}_B}$$

To calculate the risk ratio in this example:

	Low dose	High dose
No adverse events	1320	1310
Stroke, MI, or death	75	99
Total	1395	1409

The risk of stroke, MI, or death in the low-dose group is $75/1395 = 0.0538$ (or 5.38%)

The risk of stroke, MI, or death in the high-dose group is $99/1409 = 0.0702$ (or 7.02%)

The risk ratio is the ratio of the risk in the high-dose group to the risk in the low-dose group:
 $0.0702/0.0538 = 1.30$

- ❖ $RR < 1$ means risk is < reference group
- ❖ $RR = 1$ means risk is same as reference
- ❖ $RR > 1$ means the risk > reference group

Odds

Number of times outcome occurs divided by number of times it doesn't.

$$\text{Odds} = \frac{\# \text{favourable}}{\# \text{unfavourable}}$$

The odds ratio of a stroke, MI, or death for high-dose patients compared to low-dose patients is: $OR = (99/1310)/(75/1320) = 1.33$

$$OR = \frac{Odds_A}{Odds_B}$$

- ❖ Odds < 1 means outcome occurs < ½ the time
- ❖ Odds = 1 means outcome occurs ½ the time
- ❖ Odds > 1 means outcome occurs > ½ the time

Blog

Incidence and prevalence are an important concept in epidemiology that provide valuable information about the distribution and burden of a particular health outcome in a population. Especially prevalence measures include the time period. I only applied standardisations via software before, and they are calculated automatically. It is good to know how they are calculated and the difference between direct standardisation and indirect standardisation. Rates, risks and odds are easily confused. I think that is why most of the questions are relevant about them. These questions help to deepen the understanding via practice distinguish and calculation.

I think the expression of the title and options for a few questions is ambiguous, and I think one of the questions has no correct answer. I don't know if the teacher will review the questions or if there is any way to ask the teacher.

The R part continues to practice reading CSV files and editing tables and begins to practice simple calculations on the data in the table. Although it is a bit boring to be full of numbers without plots, the calculation content is very relevant to the epidemiological concepts learned in this chapter.

Chapter 4 – Statistical Computing

Descriptive statistics are used to summarize and describe the main characteristics of a dataset. The following are some commonly used descriptive concepts:

1. Mean: The mean is the arithmetic average of a set of values. It is calculated by summing up all the values and dividing by the number of values.

The mean of a vector or data frame can be calculated by using the **mean()** function. For example, if you have a vector **x**, you can calculate its mean as follows:

```
x <- c(1, 2, 3, 4, 5)
mean(x)
```

Output: 3

2. Standard deviation: The standard deviation measures the spread or dispersion of the values around the mean. It is calculated by taking the square root of the variance.

```
sd(x)
```

Output: 1.58113883008419

3. Variance: The variance is a measure of how spread out the values are in a dataset. It is calculated by taking the average of the squared differences from the mean.

```
var(x)
```

Output: 2.5

4. Covariance: The covariance is a measure of the joint variability of two variables. It indicates how two variables are related to each other. A positive covariance indicates that the variables move together, while a negative covariance indicates that the variables move in opposite directions. It is calculated by taking the average of the product of the differences of the values from their respective means.

You can calculate the covariance between two vectors using the **cov()** function. For example, if you have two vectors **x** and **y**, you can calculate their covariance as follows:

```
y <- c(2, 4, 6, 8, 10)
cov(x, y)
```

Output: 5

5. Correlation: Correlation is a measure of the strength and direction of the linear relationship between two variables. It ranges from -1 to 1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and 1 indicates a perfect positive correlation. Correlation is calculated by dividing the covariance by the product of the standard deviations of the two variables.

```
cor(x, y)
```

Output: 1

*Note: [correlation is not causation](#)

*Note: Covariance can be positive, negative, or zero. It is depending on the relationship between the variables being measured. If the greater values of one variable mainly correspond with the greater values of the other variable, and the same holds for the lesser values (that is, the variables tend to show similar behavior), the covariance is positive. When the greater values of one variable mainly correspond to the fewer values of the other, (that is, the variables tend to show opposite behavior),

the covariance is negative. When the covariance between two variables is 0, it means that there is no linear relationship between them. It's worth to note that a covariance of 0 does not necessarily mean that there is no relationship between the variables at all, but rather that any relationship that exists is not linear. It's possible that there may be a non-linear relationship between the variables, or that there may be a relationship that is not captured by the covariance measure.

Linear transformations

Linear transformations are mathematical operations that can be applied to a set of data to create a new set of transformed data. These transformations involve scaling, shifting, or a combination of both, and can be used to standardize data or convert data to a different unit of measurement.

When a linear transformation is applied to a set of data, it has an effect on both the mean and variance of the data. Specifically, if we apply a linear transformation to a set of data, where the transformation is of the form:

$$Y = aX + b$$

where Y is the transformed variable, X is the original variable, a is a scaling factor, and b is a shifting factor. Then the mean and variance of the transformed variable Y can be calculated as follows:

$$\text{Mean}(Y) = a \cdot \text{Mean}(X) + b$$

$$\text{Variance}(Y) = a^2 \cdot \text{Variance}(X)$$

From these formulas, we can see that applying a linear transformation to a set of data will change both the mean and variance of the data. Specifically, the mean will be shifted by b, and the variance will be scaled by the square of a.

For example, if we have a set of data with a mean of 50 and a variance of 25, and we apply a linear transformation of $Y = 2X + 10$, then the mean of the transformed data will be 110 ($250 + 10$), and the variance will be 100 ($2^2 \cdot 25$).

In summary, linear transformations have a predictable effect on the mean and variance of a set of data. By understanding this relationship, we can use linear transformations to standardize data or convert data to a different unit of measurement.

In case of C4_P3 exercise 3.3

As can be seen in these histograms, the age distribution of males and females is basically the same. The average height of men is higher than that of women, and the average weight and waist circumference are also higher than those of women, but there is no significant difference in height. Therefore, the BMI of men is also slightly higher than that of women. In addition, these three data are typically normally distributed with bell-shaped curves.

The histogram of the ear length is roughly bell-shaped and the points on the normal probability map are basically along a straight line. It can be judged that the ear length can be regarded as a normal distribution. In comparison, the age distribution does not conform to the normal distribution. However, it's worth noting that the normality assumption is not always necessary for statistical analysis, especially if the sample size is large enough. In some cases, it may be more appropriate to use non-parametric methods or transformations to analyze the data.

These observations suggest that there are sex differences in body data that could be further investigated.

R codes used in this cases:

plot() is a basic function in R used to create a wide range of graphs and visualizations. It is a versatile function that can be used to create scatterplots, line charts, bar charts, histograms, boxplots, and many other types of charts.

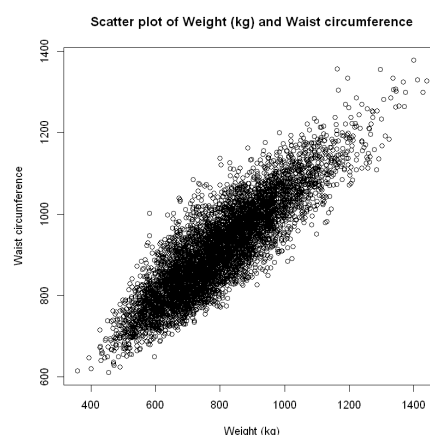
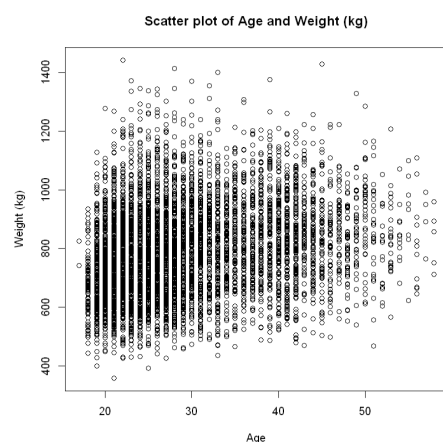
The **plot()** function takes in one or more variables as inputs, which can be numeric or categorical, and creates a visual representation of the data. The function also allows you to customize various aspects of the plot, such as the labels, colors, and axes.

Here's an example of using **plot()** to create a scatterplot:

```
# create a dataset of two variables
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 6, 8, 10)

# create a scatterplot of x vs y
plot(x, y, main = "Scatterplot of x vs y", xlab = "x", ylab = "y")
```

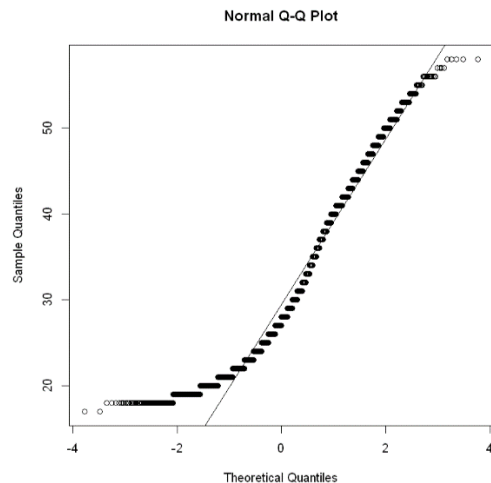
This will create a scatterplot of the **x** and **y** variables with **x** on the x-axis and **y** on the y-axis, and a title and axis labels added using the **main** and **xlab** and **ylab** arguments.



From the plots of C4_P3 It can be clearly seen that age has no correlation with weight. In these scatterplots, the data points are evenly spaced. However, a strong (but not perfect) positive correlation can be found between weight and waist circumference as well as MBI and waist circumference.

qqnorm() is used to create a normal [quantile-quantile \(QQ\) plot](#), which compares the distribution of the data to a normal distribution. It plots the quantiles of the data against the quantiles of a theoretical normal distribution on a scatter plot. If the data is normally distributed, the points on the QQ plot should follow a straight line.

qqline() is used to add a reference line to the QQ plot. This line is typically a straight line that goes through the first and third quartiles of the data. If the points on the QQ plot deviate significantly from this line, it suggests that the data is not normally distributed.



Together, `qqnorm()` and `qqline()` can be used to visually assess whether a dataset is normally distributed. If the points on the QQ plot follow a straight line that closely matches the reference line, it suggests that the data is normally distributed. If the points deviate significantly from the reference line, it suggests that the data is not normally distributed.

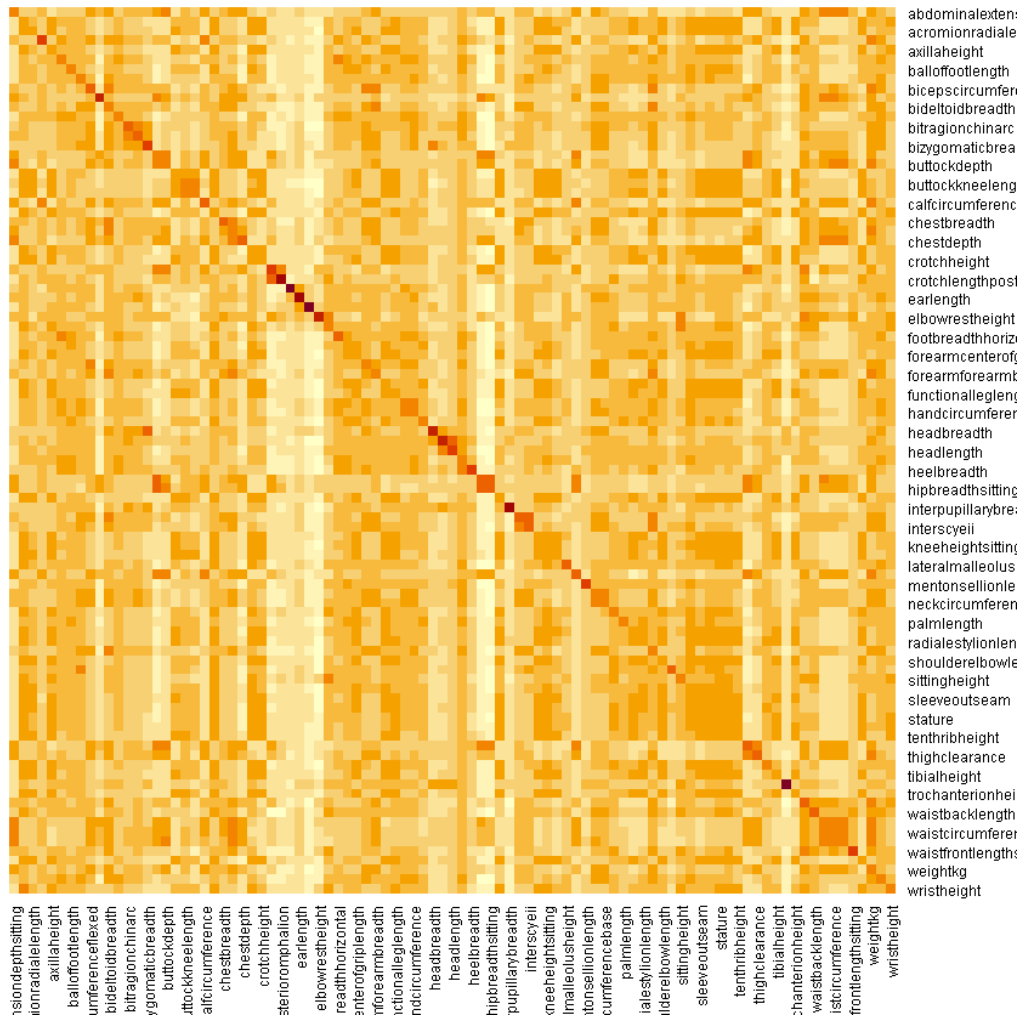
In addition for the analysis of C4_P3 using the `cor()` function can calculate the correlation matrix for the selected numeric columns. Then, the code visualizes the correlation matrix using the `heatmap()` function, which creates a heatmap with color-coded cells representing the strength and direction of the correlation between each pair of variables.

```
# Select only numeric measurement columns
# In R, columns are indexed starting from 1, so adata[,2:94] selects all rows and columns 2
# through 94.
numeric_data <- adata[,2:94]

# Calculate correlation matrix
correlation_matrix <- cor(numeric_data)

# Visualize correlation matrix
heatmap(correlation_matrix, Rowv=NA, Colv=NA, revC=TRUE)
```

HDAT9200 Statistical Foundations for Health Data Science Course Reflection including collation of Phase 4 (chapter reflection and blog) posts



Blog

Covariance is quite a new concept for me; thus, I select it as my MCQ topic. The difference between Correlation and Regression is also important but not introduce in this Chapter, [here](#) is a short instruction chart with definition & comparison

Chapter 5 – Introduction to Probability Theory

1. Kolmogorov's three axioms of probability

Axiom 1 Non-Negativity: For any event A , $P(A) \geq 0$

The smallest value for $P(A)$ is zero and if $P(A)=0$, then the event A will never happen.

Axiom 2 Normalization: Probability of the entire sample space S is equal to 1; $P(S)=1$

the probability of the whole sample space is equal to one, i.e., 100 percent.

Axiom 3 Additivity: The probability of the union of two disjoint events is equal to the sum of their individual probabilities. That is if A_1, A_2, A_3, \dots are disjoint events, then $P(A_1 \cup A_2 \cup A_3 \dots) = P(A_1) + P(A_2) + P(A_3) + \dots$

The basic idea is that if some events are disjointed (i.e., there is no overlap between them), then the probability of their union must be the summations of their probabilities. Another way to think about this is to imagine the probability of a set as the area of that set in the Venn diagram.

*It's an interesting idea to think of probability as an area: For example, in probability density functions, the area under the curve between two points represents the probability that a random variable fall between those two points. The total area under the curve is equal to 1, since the probability of any possible outcome must be between 0 and 1, and the sum of all possible outcomes must be equal to 1.

Notation: $P(\emptyset) = 0$

If $A \subseteq B$ then $P(A) \leq P(B)$

$P(A \cap B) = P(A \text{ and } B) = P(A, B)$

$P(A \cup B) = P(A \text{ or } B) = P(A) + P(B) - P(A \cap B)$

$P(A^c) = P(\Omega \setminus A) = 1 - P(A)$

2. The Law of Large Numbers (LLN)

In simple terms, as the sample size increases, the sample mean approaches the population mean.

- ❖ The weak law: For any small positive number, epsilon, as the sample size increases, the probability that the absolute difference between the sample mean and the population mean is greater than epsilon approaches zero. This means that as the sample size increases, the sample mean becomes a more and more accurate estimate of the population mean.
- ❖ The strong law: the sample mean converges to the population mean almost surely. This means that for any possible outcome, the sample mean approaches the population mean.

3. The Central Limit Theorem (CLT)

[Central Limit Theorem \(mfviz.com\)](https://mfviz.com)

The central limit theorem (CLT) states that, given certain conditions, the arithmetic mean of a sufficiently large number of iterates of independent random variables, each with a well-defined (finite) expected value and finite variance, will be approximately normally distributed, regardless of

the underlying distribution. In addition, the standard deviation of the sample means will approach the standard error of the mean.

- It is often used to construct confidence intervals and perform hypothesis testing.
- It allows us to use the normal distribution as an approximation for many real-world problems, even when the underlying distribution is not normal.
- It may not hold for certain types of distributions with heavy tails, such as the Cauchy distribution.
- It does not apply to medians or other non-parametric statistics.

4. R codes

4.1 The **sample()** function in R is used to generate random samples of a specified size from a given set of values or objects. It has the following syntax:

```
sample(x, size, replace = FALSE, prob = NULL)
```

- **x** is a vector of values or objects from which to generate the random samples.
- **size** is the number of random samples to generate.
- **replace** is a logical value indicating whether or not to sample with replacement. If **replace = TRUE**, the same value can be sampled more than once.
- **prob** is a vector of probabilities associated with each element of **x**.

If **replace = TRUE**, **prob** can be used to specify the probability of each element being selected. If **replace = FALSE**, **prob** is ignored and each element has an equal probability of being selected.

4.2 A coin tossing engine

```
toss_coin <- function(tosses, proportions=FALSE, ...) {
  x <- c("H", "T")
  tosses <- sample(x, size=tosses, replace=TRUE, ...)
  tosses_table <- table(tosses)
  if (proportions) {
    return(prop.table(tosses_table))
  } else {
    return(tosses_table)
  }
}
```

The **toss_coin** function in R simulates the tossing of a coin a specified number of times and returns the counts or proportions of heads and tails. Here's what the function does:

- The function takes two arguments: **tosses** and **proportions**, with **tosses** being the number of times to simulate the coin toss and **proportions** being a logical value indicating whether the function should return proportions (default is counts).
- The function defines a character vector **x** with two elements, "H" and "T", representing heads and tails, respectively.

- The `sample()` function is used to simulate the tossing of the coin tosses times. The `size` argument specifies the number of samples to take, and `replace=TRUE` allows for sampling with replacement, meaning that the same outcome (heads or tails) can occur multiple times.
- The function uses the `table()` function to create a frequency table of the outcomes, counting the number of times heads and tails occur.
- If `proportions` is set to `TRUE`, the function uses the `prop.table()` function to return the proportion of times heads and tails occur. If `proportions` is `FALSE`, the function returns the frequency table of the outcomes.

```
for (i in seq_along(a$tosses)) {
  a[i, "prop_heads"]<- toss_coin(a[i, "tosses"], prop=TRUE)[[1]]
}
```

This is a for loop that iterates over the numbers in the `tosses` column of the data frame `a`. For each value of `tosses`, the `toss_coin()` function is called with `prop=TRUE` to simulate a coin toss and calculate the proportion of heads. The resulting proportion is stored in the `prop_heads` column of the corresponding row in a data frame.

4.3 Customized Function

```
# define means_of_samples function
means_of_samples <- function(data, sample_size, number_of_samples) {
  means <- numeric(number_of_samples)
  for (i in 1:number_of_samples) {
    s <- sample(data, sample_size, replace=TRUE)
    means[i] <- mean(s)
  }
  return(means)
}
```

The

`means_of_samples()` function defined here takes a dataset, a sample size, and the number of samples to be drawn from the dataset. It then generates the means of the specified number of samples of the specified sample size and returns these sample means.

```
# define hno function
hno <- function(sample_means, ...) {
  hist(sample_means, breaks=50, prob=TRUE, ...)
  x <- seq(min(sample_means), max(sample_means), length.out=1000)
  curve(dnorm(x, mean=mean(sample_means), sd=sd(sample_means)), col = "red", add = TRUE)
}
```

The `hno()` function defined here takes a vector of sample means and generates a histogram of the sample means with the specified number of breaks. It also adds a density curve to the histogram using the `dnorm()` function with the mean and standard deviation of the sample means.

5. MCQ discussion

I have question about Caitlin Tjoa's MCQ in C5

A PCR test has been developed to diagnose COVID, which affects 1 in 1000 people. The PCR is 99% accurate; this means it correctly identifies 99% of people who have COVID and 99% of people who do not. If a patient tests positive for COVID, what is the probability that they truly do have COVID? Use this calculation to determine which of the following statements is FALSE according to Kolmogorov's three axioms of probability.

1. The first axiom stating all probabilities are non-negative, holds true.

HDAT9200 Statistical Foundations for Health Data Science
Course Reflection including collation of Phase 4 (chapter reflection and blog) posts

2. The second axiom stating that the probability of the entire sample space is 1, holds true.
3. The third axiom, which is the assumption of additivity, holds true.
4. One of the axioms does not hold true in this scenario.

I select 4. however, the feedback shows 3 is correct answer.

Here is the explain that Caitlin Tjoa given:"

This is incorrect because this is a true statement. The third axiom does not hold true. The probability of a patient having COVID is 1 in 1000 or 0.1% therefore the probability of not having COVID is 99.9%. The PCR will correctly identify the 1 person with COVID (true positive) but will incorrectly identify 1% of the 999 people as positive (false positive). Out of the 2 people that will test positive, only 1 actually has COVID. So, the probability that the patient actually has COVID given a positive result is $1/2$ or 50%. This violates the third axiom because the events "patient has COVID and tests positive" and "patient does not have COVID and tests positive" does not add up to 1."

I think the explanation provided for the third axiom is incorrect. The third axiom, the assumption of additivity, states that the probability of the union of two events is equal to the sum of their individual probabilities minus the probability of their intersection. It does not state that the probability of all events must add up to 1. In the given scenario, the events are "patient has COVID and tests positive" and "patient does not have COVID and tests positive", and their intersection is the event "patient does not have COVID but tests positive" which has a probability of $0.01 \times 999/1000 = 0.00999$. Therefore, the probability of the union of the two events is $1/1000 + 0.99 \times 0.01 = 0.0199$, and the probability that the patient actually has COVID given a positive result is $1/20$ or 5%. This does not violate the third axiom.

Andrew's Answer:

I agree with you, the author of this particular questions appears to have got themselves in a confusion over conditional probabilities.

If the outcome is COVID +ve or -ve, then the probabilities are 0.001 & 0.999, respectively. A sum of 1.

Taking the definition of accuracy as stated, then the probability of being +ve given tested +ve (as asked in the question stem), is the accuracy. That is, 0.99 (& 0.01 that someone who tests +ve is truly -ve). Again, a sum of one. The exact same holds for being truly -ve if tested -ve.

The PCR will correctly identify the 1 person with COVID (true positive) but will incorrectly identify 1% of the 999 people as positive (false positive). Out of the 2 people that will test positive, only 1 actually has COVID. So, the probability that the patient actually has COVID given a positive result is $1/2$ or 50%.

Is not correct. The accuracy (as defined in the question stem) says that 0.01 of all tests (whether +ve or -ve) will be incorrect. Given the prevalence, we can determine that per 100,000 tests there would be 100 +ve's & 99900 -ve's. Then given the accuracy, we can determine:

- 99 true +ve's
- 1 false -ve
- 999 false +ve's
- 98901 true -ve's

However, none of this gets us any closer to the $1/2 = 50\%$ as stated in the feedback. When considering these conditional probabilities as the outcome, there are 4 'states' (not 2) in the outcome space. Thus, when considering the additivity axiom, we need to consider all 4 independent states (not 2). The above numbers would still yield probabilities for these 4 states that sum to 1.

"patient has COVID and tests positive" and "patient does not have COVID and tests positive", and their intersection is the event "patient does not have COVID but tests positive"

correct. However:

- $P(+ve \mid +ve) = 99 / 1098$
- $P(-ve \mid +ve) = 999 / 1098$

Of course, we're talking about sensitivity, specificity, PPV, NPV, & accuracy here.

[Blog](#)

<Optional: Copy-and-paste any optional Phase 4 Blog posts here>

Chapter 6 – Probability Distribution Functions

1. The Normal distribution

```
# Set the random seed
seed <- 55555

# Define the is.normal function
is.normal <- function(size) {
  # Set the random seed
  set.seed(seed)

  # Create a 3x4 plotting pane
  par(mfrow=c(3,4))

  # Repeat 12 times
  for (i in 1:12) {
    # Draw a random sample of size from a standard Normal distribution
    draw <- rnorm(size)

    # Plot a histogram of the density, with xlim=c(-3,3) and ylim=c(0,1)
    hist(draw, prob=TRUE, xlim=c(-3,3), ylim=c(0,1), main=paste("Sample size =", size, ",
Plot", i))

    # Add an overlay of the density as a line
    x <- seq(-3, 3, length.out=1000)
    lines(x, dnorm(x), col="red", lwd=2)
  }
}

# Call the is.normal function with different sample sizes
is.normal(10)
is.normal(30)
is.normal(50)
is.normal(100)
is.normal(200)
is.normal(1000)
```

Observations:

- As the sample size increases, the histograms become smoother and more bell-shaped.
- The density lines become more closely aligned with the histograms as the sample size increases.
- The histograms are more variable and have more skewness at smaller sample sizes.
- The shape of the histograms becomes more normal-like as the sample size increases, even for the smallest sample size of 10.
- The overall pattern across sample sizes is that the sample means tend to be normally distributed, as predicted by the Central Limit Theorem (Chapter 5).

2. [Probability density function \(PDF\)](#)

The probability density function (PDF) is a function that describes the relative likelihood of a random variable taking on a certain value or range of values. The PDF is used to calculate the

probability that a random variable falls within a particular range of values. The area under the PDF within a given range is the probability that the random variable falls within that range.

For a continuous random variable, the PDF is a smooth function that describes the probability of the variable taking on any particular value within a range. As mentioned in Chapter 5 Kolmogorov's three axioms of probability, the PDF is normalised so that the total area under the curve is equal to 1. The height of the PDF at any point represents the probability density at that point. In addition, the Central Limit Theorem states that as the sample size increases, the distribution of the sample mean approaches a normal distribution regardless of the shape of the population distribution. This means that if we take many samples of the same size from any population, the distribution of sample means will be approximately normal.

```
set.seed(seed)
norm.pdf <- function(my_mu, my_sd ) {
  lims <- c(my_mu - 3 * my_sd, my_mu + 3 * my_sd )
  space <- seq(lims[1], lims[2], 0.01 )
  calc1 <- my_sd * sqrt(2 * pi)
  calc2 <- (space - my_mu )**2
  calc3 <- 2 * my_sd**2
  calc4 <- 1/calc1 * exp(- calc2 / calc3 )
  return(calc4)
}

empir <- rnorm(10000, 0, 1 )
pdf.est <- norm.pdf(0, 1 )
span <- seq(-3, 3, 0.01 )
plot(span, pdf.est, type="l", col="blue")
lines(density(empir), col="red")
```

- defines a function called `norm.pdf` which takes two arguments: `my_mu`, representing the mean of the normal distribution, and `my_sd`, representing the standard deviation. This function returns a vector of values representing the probability density function (PDF) of the normal distribution with the specified mean and standard deviation.
- The function calculates the PDF for the normal distribution by first creating a sequence of values ranging from `my_mu - 3*my_sd` to `my_mu + 3*my_sd` in increments of 0.01, using the `seq()` function. The `calc1`, `calc2`, `calc3`, and `calc4` variables are intermediate calculations for the formula of the normal distribution PDF. The final calculation returns the PDF values for each point in the `space` vector.
- Using the `rnorm()` function to generates two random samples from a standard normal distribution with mean 0 and standard deviation 1.
- For each sample, the code estimates the PDF of the distribution by calling the `norm.pdf()` function with mean 0 and standard deviation 1, and then plots the estimated PDF as a blue line using the `plot()` function. The code then overlays the density estimate of the sample as a red line using the `lines()` function.

When comparing PDFs, we can look at their shape, central tendency, and spread. One way to visualize the comparison of PDFs is to plot them on the same graph. Here are some scenarios that might arise when comparing PDFs:

- If two PDFs have the same shape, they have the same central tendency and spread. The only difference between them is their scaling. In this case, one PDF is a scaled version of the other.
- If two PDFs have different shapes but the same mean and standard deviation, they have the same central tendency and spread. The difference between them is the way the probability is distributed around the mean.

- If two PDFs have different shapes and different means, they have different central tendencies. We can compare the relative heights of their peaks to see which distribution is higher or more likely.
- If two PDFs have different shapes and different variances, they have different spreads. We can compare the widths of their peaks to see which distribution is more spread out.

3. [Cumulative distribution function \(CDF\)](#)

The cumulative distribution function (CDF) of a random variable is the probability that the variable takes a value less than or equal to a given value. The CDF of a standard normal distribution is denoted by $\Phi(z)$, where z is a random variable that follows a standard normal distribution.

Suppose in a given sample of 223 patients, the mean total cholesterol level is 11.1 millimoles per litre (mmol/L), and standard deviation 1.1 mmol/L. A total cholesterol level below 11.1 mmol/L is considered a **good** thing.

In R, the `pnorm()` function is used to calculate the cumulative probability of a standard normal distribution up to a given value or a range of values.

```
pnorm(13.3, mean=11.1, sd=1.1, lower.tail=TRUE/FALSE)
```

- The first argument 13.3 specifies the value of interest or cutoff, in this case the cutoff for high cholesterol.
- The second argument mean=11.1 specifies the mean of the normal distribution, which is the sample mean total cholesterol level of 11.1 mmol/L.
- The third argument sd=1.1 specifies the standard deviation of the normal distribution, which is the sample standard deviation of 1.1 mmol/L.
- The lower.tail=FALSE argument is used to indicate that we are interested in the upper tail of the distribution, i.e. the proportion of the sample with a total cholesterol level greater than or equal to 13.3 mmol/L.
- If lower.tail were set to TRUE, then `pnorm()` would return the proportion of the sample with a total cholesterol level less than or equal to 13.3 mmol/L.

The R function **`ecdf()`** computes an empirical cumulative distribution function

```
# set the random seed
set.seed(55555)

# generate chol
chol <- rnorm(223, 11.1, 1.1)

# plot the cumulative distribution of chol
plot(ecdf(chol), main="Cumulative Distribution of Cholesterol", xlab="Cholesterol",
     ylab="Cumulative Probability")
```

This will generate a plot of the empirical cumulative distribution of chol, showing how the cholesterol values are distributed across the range of possible values. The x-axis represents the cholesterol values and the y-axis represents the cumulative probability of observing a value less than or equal to the corresponding x-value.

For the normal distributions, the PDF is a bell-shaped curve that shows the probability density of the variable at different values. The CDF is a cumulative curve that shows the probability of the variable being less than or equal to a certain value. The mean and standard deviation of a normal distribution determine the center and spread of the distribution, respectively.

CDF is determined by the standard deviation, with a smaller standard deviation resulting in a narrower curve / increases more rapidly.

One option to help assess if a sample is drawn from a particular distribution is quantile-quantile or Q-Q plots. (already reviewed in **case of C4_P3 exercise 3.3**)

A Normal Q-Q plot plots the quartiles of your data against the quartiles of a theoretical Normal distribution. It then stands to reason that if the sample follows a Normal distribution, it should 'mimic' the behaviour of the theoretical quantiles. If the plots deviate from the straight line, especially in the tails, indicating that those samples may not be perfectly Normal.

Comparing `norm.qq(20)` and `norm.qq(200)`, we can observe that as the sample size increases, the Q-Q plots become more tightly clustered around the straight line, indicating that the sample is more likely to be drawn from a Normal distribution. This is consistent with the Central Limit Theorem, which states that as the sample size increases, the distribution of the sample mean becomes more Normal.

4. The Binomial distribution

1. Described by two parameters, n the number of trials and p the probability of success for each trial
2. Trials are independent
3. The Bernoulli distribution is a special case of the Binomial distribution with a single trial

```
norm.bin <- function(num.trials, p) {  
  
  # Generate num.obs random samples from the binomial distribution  
  num.obs <- 100000  
  bin.draw <- rbinom(num.obs, size=num.trials, prob=p)  
  
  # Calculate the mean and standard deviation of the normal distribution  
  mu <- num.trials * p  
  sigma <- sqrt(num.trials * p * (1 - p))  
  
  # Generate num.obs random samples from the normal distribution  
  norm.draw <- rnorm(num.obs, mean=mu, sd=sigma)  
  
  # Create a histogram of the binomial distribution with a blue border  
  hist(bin.draw, border="blue")  
  
  # Add a histogram of the normal approximation to the binomial distribution with a red  
  border  
  hist(norm.draw, border="red", add=T)  
}
```

This function generates a histogram of a binomial distribution with parameters size (i.e., the number of trials) and prob (i.e., the probability of success), as well as a histogram of a normal approximation to the binomial distribution using the central limit theorem. The function takes two arguments: `num.trials`, the number of trials for the binomial distribution, and `p`, the probability of success for the binomial distribution.

The Normal distribution approximates to the Binomial distribution for any p , provided n is sufficiently large (n increases indefinitely). This approximation forms quicker for p closer to 0.5, since the symmetry of the Binomial distribution increases as p tends to 0.5.

So as n increases indefinitely, the Normal approximates to the Binomial with **mean = np** and **standard deviation = $\sqrt{np(1-p)}$**

`rbinom()` is a function in R that generates random numbers from a binomial distribution. The function takes three arguments:

- `n`: the number of random values to generate.
- `size`: the number of trials in the binomial experiment.
- `prob`: the probability of success in each trial.

The function returns a vector of `n` random values generated from the binomial distribution with specified parameters. For example, `rbinom(10, 5, 0.5)` generates 10 random values from a binomial distribution with 5 trials and a probability of success of 0.5.

5. [The Poisson distribution](#)


The Poisson distribution is a discrete probability distribution that expresses the probability of a given number of events occurring in a fixed interval of time or space, given that these events occur with a known constant rate and independently of the time since the last event.

The Poisson distribution is used to model rare events, such as the number of car accidents per day in a city, the number of calls to a customer service center in an hour, or the number of particles emitted from a radioactive source in a given time interval.

The probability mass function (PMF) of the Poisson distribution is given by:

$$P(X = k) = (\lambda^k * e^{(-\lambda)}) / k!$$

where λ is the mean number of events in the interval and k is the number of events that occur.

The Poisson distribution has some  unique properties:

- The mean and variance of the Poisson distribution are both equal to λ .
- The Poisson distribution is memoryless, meaning that the probability of an event occurring in a given time interval is independent of the time elapsed since the last event.
- The Poisson distribution approaches the normal distribution as the mean λ becomes large.
- The Normal distribution is also a limiting case of the Poisson distribution under the condition that $\mu \rightarrow \infty$.

Blog

<Optional: Copy-and-paste any optional Phase 4 Blog posts here>

Chapter 7 – Likelihood

Likelihood and Maximum Likelihood Estimation

Great explain on YouTube:

[Maximum Likelihood, clearly explained!!!](#)

[Maximum Likelihood For the Normal Distribution, step-by-step!!!](#)

[Maximum Likelihood for the Binomial Distribution, Clearly Explained!!!](#)

[Probability is not Likelihood. Find out why!!!](#)

[Maximum Likelihood estimation - an introduction part 1](#)

[Maximum Likelihood estimation - an introduction part 2](#)

Probability

probability is the chance of an event occurring, while likelihood is the probability of the observed data given a particular parameter value. Probability is used to make predictions about future events, while likelihood is used to estimate the parameters of a statistical model based on observed data.

- The prior probability is the predicted probability based on background common sense or statistics of historical data, and only contains one variable, such as $P(X)$, $P(Y)$.
- Conditional probability is the probability that an event occurs after another event occurs, for example, $P(X|Y)$ represents the probability that event Y occurs after event X occurs.
- The posterior probability is to find the cause by the effect, the probability of finding the cause when the result is known. For example, event Y is caused by X , then $P(X|Y)$ is the posterior probability, or it can be said that it is the reverse conditional probability after the occurrence of the event.

The differences between probability and likelihood

- probability refers to the chance of an event happening before the data is observed, while likelihood refers to the chance of the observed data happening given a particular parameter value.
- probability is used to make predictions about future events, while likelihood is used to estimate the parameters of a statistical model based on observed data.

Likelihood

Likelihood is a concept in statistics that refers to the probability of observing a set of data given a particular value or set of values for the parameters of a statistical model. In other words, it measures how well a particular set of parameters fit the observed data. The likelihood function is defined as the probability of the observed data given the parameter values of a statistical model.

The likelihood function can be written (in R) as:

```
lik_bern <- function(p, draw ) {  
  p^sum(draw==1) * ( ( 1 - p )^sum(draw==0) )  
}
```

This is a likelihood function for the Bernoulli distribution, which takes a probability parameter p and a vector of 0's and 1's $draw$ as input. It computes the likelihood of the data given the parameter p using the formula: $p^{\sum(draw==1)} * ((1 - p)^{\sum(draw==0)})$

Here, $\text{sum}(\text{draw}==1)$ counts the number of 1's in the data, which represents the number of successes, and $\text{sum}(\text{draw}==0)$ counts the number of 0's in the data, which represents the number of failures. The likelihood function computes the probability of observing the data given the parameter p under the assumption that the data follows a Bernoulli distribution with parameter p .

The likelihood function can be used to estimate the parameter p using maximum likelihood estimation (MLE), which involves finding the value of p that maximizes the likelihood function. The MLE of p is the value that makes the observed data most likely, given the assumed distribution.

Maximum Likelihood Estimation

Maximum likelihood estimation is the first and most natural application of the likelihood function. The maximum value of the likelihood function indicates that the corresponding parameters can make the statistical model the most reasonable. Starting from such an idea, the method of maximum likelihood estimation is:

- first select the likelihood function (usually the probability density function or probability mass function),
- then find the maximum value after sorting. In practical applications
- the logarithm of the likelihood function is generally used as the function for finding the maximum value, so that the obtained maximum value is the same as the result obtained by directly calculating the maximum value. The maximum value of the likelihood function is not necessarily unique, nor does it necessarily exist.

Maximum Posteriori Probability (MAP)

$$P(\alpha|X) = \frac{P(X|\alpha)P(\alpha)}{P(X)}$$

Among them, $P(\alpha|X)$ on the left side of the equation represents the posterior probability, and the optimization goal is $\text{argmax}_{\alpha} P(\alpha|X)$, the probability of the model parameter α appearing is maximised after the observation value X is given. The molecular formula $P(X|\alpha)$ on the right side of the equation is the likelihood function $L(\alpha|X)$.

MAP considers the prior probability $P(\alpha)$ of the occurrence of the model parameter α . Even if the likelihood is very large, but the possibility of α appearing is tiny, and it is more inclined not to consider the model parameter as α .

R codes studied in this chapter:

1. The function **dbinom()** is used to compute the probability mass function (PMF) of a binomial distribution. The binomial distribution is a discrete probability distribution that represents the number of successes in a fixed number of independent trials, where each trial has the same probability of success.

The **dbinom()** function takes the following arguments:

- **x**: the number of successes
- **size**: the number of trials
- **prob**: the probability of success in each trial

The function returns the probability of getting exactly **x** successes in **size** trials, given a probability of success of **prob**.

2. The **dnorm()** function in R is used to compute the probability density function (PDF) of a normal distribution. It takes three arguments:

1. **x**: the value(s) at which to evaluate the PDF
2. **mean**: the mean of the normal distribution
3. **sd**: the standard deviation of the normal distribution

The function returns the PDF value(s) for the given input(s) **x**.

Here's an example usage of **dnorm()**:

3. The **mle()** function in R is used to perform maximum likelihood estimation for a user-defined likelihood function. It returns the maximum likelihood estimate of the parameters for a given model. The input arguments to the **mle()** function are:

- **fn**: A user-defined likelihood function that takes in the parameters to be estimated as input and returns the log-likelihood of the data given those parameters.
- **start**: A named list of starting values for the parameters.
- **method**: The optimization method to be used. By default, the **optim()** function is used, but other optimization methods such as Nelder-Mead and Broyden-Fletcher-Goldfarb-Shanno (BFGS) can also be specified.
- **...**: Additional arguments that are passed to the optimizer.

The **mle()** function returns an object of class "mle" that contains the maximum likelihood estimates, standard errors, log-likelihood, and other information about the optimization process.

Here's an example of how to use the **mle()** function to estimate the parameters of a binomial distribution:

```
library(stats4)
dbinom(3, size=10, prob=0.5 )

# Set random seed
set.seed(88888)

# Generate some sample data from a normal distribution
x <- rnorm(100, mean = 5, sd = 2)

# Define the log-likelihood function for a normal distribution
loglik <- function(mu, sigma) {
  -sum(dnorm(x, mean = mu, sd = sigma, log = TRUE))
}

# Estimate the parameters using maximum likelihood estimation
fit <- mle(loglik, start = list(mu = mean(x), sigma = sd(x)), method = "L-BFGS-B", lower = c(-Inf, 0), upper = c(Inf, Inf))

# Print the estimated parameters
summary(fit)
```

This code performs maximum likelihood estimation to estimate the parameters of a normal distribution that generated some sample data.

- The function **dbinom()** is used to calculate the probability of a certain number of successes (3) in a certain number of Bernoulli trials (10) with a given probability of success (0.5).
- The code then sets a random seed using **set.seed()**, generates a sample of 100 random numbers from a normal distribution with mean 5 and standard deviation 2 using **rnorm()**, and stores the sample in **x**.

- The `loglik()` function is defined to calculate the negative log-likelihood of a normal distribution with mean μ and standard deviation σ given the sample x . The negative log-likelihood is used instead of the log-likelihood because the optimization algorithms used by `mle()` aim to minimize the function rather than maximize it.
- The `mle()` function is then called with `loglik` as the log-likelihood function, `start` as the starting values for the parameters (the sample mean and standard deviation), `method` as the optimization algorithm to use (in this case, "L-BFGS-B" is a quasi-Newton method for bounded optimization), and `lower` and `upper` as the lower and upper bounds on the parameter values.
- Finally, the estimated parameters are printed using `summary(fit)`.

Maximum Likelihood Estimation (MLE) is a method used to find the values of the parameters of a statistical model that maximize the likelihood function, given the observed data. This is done by finding the values of the parameters that make the observed data most likely to have been generated by the model. MLE is a popular method for estimating parameters in statistical models because it provides a principled way to estimate model parameters and is often relatively simple to implement.

To find the maximum likelihood estimates, one typically takes the derivative of the likelihood function with respect to the parameters, sets the derivative equal to zero, and solves for the values of the parameters that maximize the likelihood function. In some cases, the solution can be found analytically, while in other cases, numerical optimization techniques may be necessary. The maximum likelihood estimates are the parameter values that maximize the likelihood function, and these are often used as the best estimates of the true parameter values.

[Blog](#)

<Optional: Copy-and-paste any optional Phase 4 Blog posts here>

Chapter 8 – Frequentist Statistics

Inference and Hypothesis Testing

[Hypothesis testing and p-values | Inferential statistics | Probability and Statistics | Khan Academy](#)

[Hypothesis testing \(ALL YOU NEED TO KNOW!\)](#)

Inference is the process of making conclusions or predictions about a population based on information or data collected from a sample. It involves using statistical techniques to analyse and interpret data in order to make statements or draw conclusions about a larger group or population.

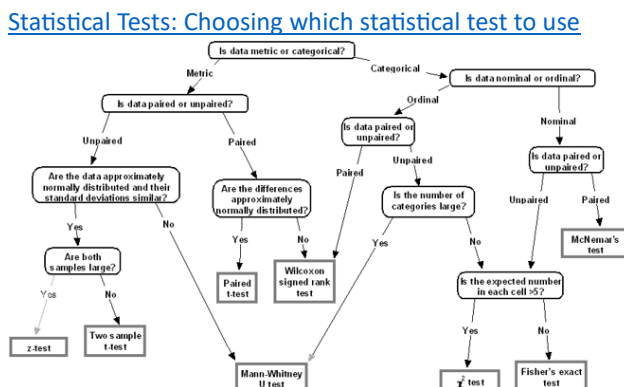
Hypothesis testing is a common inference technique in statistics. It involves making a statement, called a hypothesis, about a population parameter and then using data to determine whether there is enough evidence to support or reject the hypothesis. The process involves selecting a level of significance, which is the probability of rejecting the null hypothesis when it is actually true, and then using a test statistic and a p-value to make a decision about whether to reject or fail to reject the null hypothesis.

The two types of hypothesis testing:

- The null hypothesis (H_0) is typically a statement that assumes no difference or no relationship between variables.
- The alternative hypothesis (H_1) is the opposite of the null hypothesis.

The five stages of a hypothesis test:

1. State H_0 and H_1 hypotheses.
2. Decide level of significance.
need to decide what level of probability (α) we will accept as being 'unlikely'. Conventionally a 5% probability is considered sufficiently unlikely. i.e. $\alpha = 0.05$
3. Define and evaluate test statistic - choose a test statistic that has a distribution that matches our data.
([Choosing the Right Statistical Test | Types & Examples](#))



4. Calculate the P-value.
5. Interpret the results.
If the P-value is less than our pre-set value of α (0.05) then;
 - we have a statistically significant result.
 - we might comment that there is strong evidence against the null hypothesis of no difference.
 - we reject the null in favour of the alternative.If the p value is more than our pre-set value of α (0.05) then;
 - we have a statistically non-significant result.
 - we might comment that there is insufficient evidence for the alternative hypothesis of a difference.
 - we fail to reject the null.

P-values

Type 1 error (α) is the chance of detecting a statistically significant difference when the treatments are really equally effective.

- probability of rejecting the null when in fact it is true.
- i.e. risk of a false-positive result.

Type 2 error (β) is the chance of not detecting a significant difference when there really is a difference.

- probability of failing to reject the null, when in fact it is false.
- i.e. risk of a false-negative.

(See **Bias** in Chapter1)

Power ($=1 - \beta$) is the chance of a true-positive.

- probability of rejecting the null when in fact it is false.
 - i.e. the chance of not getting a false-negative.
 - i.e. the chance of spotting a difference as statistically significant if there really is a difference of a given size.
- the ability to detect a true difference of a given (clinical) importance.
- can be thought of as “the confidence with which the investigator can claim that a specified treatment benefit has not been overlooked”.
- Aim for 90% power, but 80% is ok

R codes:

qt() is a function that returns the critical t-value for a two-tailed test with a given probability and degrees of freedom.

The syntax of **qt()** function is: `qt(p, df, lower.tail = TRUE/ FALSE)`

- **p**: the probability for which to find the quantile
- **df**: the degrees of freedom
- **lower.tail**: logical; if **TRUE** (default), the function returns the probability that a random variable from a t-distribution is less than or equal to the quantile; if **FALSE**, it returns the probability that a random variable from a t-distribution is greater than the quantile.

t.test() is a function in R that performs a t-test.

`t.test(x, y, var.equal=TRUE)`

- x and y, which are the data sets to be compared
- If only one argument is provided, it is assumed to be the first data set (x), and the second data set (y) is assumed to be absent.
- The var.equal=TRUE argument specifies that the variance of the two samples is assumed to be equal.
- If this argument is not specified or is set to FALSE, then a Welch's t-test is performed, which does not assume equal variances.

The **chisq.test()** function in R is used for conducting a chi-squared test of independence between two categorical variables. It tests the null hypothesis that the two variables are independent (i.e., not related to each other) against the alternative hypothesis that they are dependent (i.e., related to each other).

The syntax of the **chisq.test()** function is:

HDAT9200 Statistical Foundations for Health Data Science
Course Reflection including collation of Phase 4 (chapter reflection and blog) posts

`chisq.test(x, y = NULL, correct = TRUE, p = rep(1/length(x), length(x)), rescale.p = FALSE, simulate.p.value = FALSE, B = 2000)`

- **x**: a table containing the observed counts of the two variables. This can be a matrix or a data frame.
- **y**: if **x** is a matrix or data frame, **y** can be a vector containing the row labels, or **NULL** if the rows are already named.
- **correct**: a logical value indicating whether to apply a continuity correction to the test statistic. The default is **TRUE**.
- **p**: a vector of expected proportions (under the null hypothesis). By default, this is set to the proportions of the marginal totals of the observed counts.
- **rescale.p**: a logical value indicating whether to rescale the expected proportions to sum to the same value as the observed counts. The default is **FALSE**.
- **simulate.p.value**: a logical value indicating whether to compute the p-value by Monte Carlo simulation. The default is **FALSE**.
- **B**: the number of simulations to use when computing the p-value by Monte Carlo simulation.

The `chisq.test()` function returns a list containing the following components:

- **statistic**: the value of the chi-squared test statistic.
- **parameter**: the degrees of freedom of the chi-squared distribution.
- **p.value**: the p-value of the test.
- **method**: a character string indicating the type of test performed.
- **data.name**: a character string giving the name of the data.
- Here is an example of how to use the `chisq.test()` function:

The **qnorm()** function is used to compute the quantiles of the standard normal distribution. It takes a probability value as an argument and returns the corresponding value from the standard normal distribution that has that probability. `qnorm(0.95, mean = 10, sd = 2)` would return the 95th percentile of a normal distribution with mean 10 and standard deviation 2.

Case from exercise 3.6

```
# Histograms and QQplots for all variables
par(mfrow = c(3,4)) # Set up the plotting grid

for (i in 7:11) { # Loop through columns 7 to 11 (excluding non-numeric columns)
  hist(dys[, i], main = paste0("Histogram of ", colnames(dys)[i]))
  qqnorm(dys[, i], main = paste0("QQplot of ", colnames(dys)[i]))
  qqline(dys[, i])
}
```

- This loop iterates through columns 7 to 11 of the **dys** dataset. For each column, it generates a histogram and a QQplot using the **hist()** and **qqnorm()** functions, respectively.
- The **main** argument in each function is used to set the title of the plot, which includes the name of the variable from the dataset.
- The **qqline()** function is then used to add a reference line to the QQplot.

```
# Calculate correlation matrix
pegboard_corr <- cor(dys[, c("Pegboard1", "Pegboard2", "Pegboard3", "Pegboard4", "Pegboard5")])

# Print the correlation matrix
pegboard_corr
```

	Pegboard1	Pegboard2	Pegboard3	Pegboard4	Pegboard5
Pegboard1	1.0000000	0.5881402	0.3984903	0.4602274	0.2071554
Pegboard2	0.5881402	1.0000000	0.3543970	0.1973244	0.2336377
Pegboard3	0.3984903	0.3543970	1.0000000	0.3477014	0.2708679
Pegboard4	0.4602274	0.1973244	0.3477014	1.0000000	0.4121548
Pegboard5	0.2071554	0.2336377	0.2708679	0.4121548	1.0000000

- The code you provided calculates the correlation matrix for the variables "Pegboard1", "Pegboard2", "Pegboard3", "Pegboard4", and "Pegboard5" from the data frame **dys**.
- The **cor()** function is used to calculate the correlation matrix. By default, it uses Pearson's correlation coefficient to measure the linear relationship between pairs of variables. The resulting correlation matrix is a symmetric matrix where the diagonal elements are always equal to 1 (since each variable is perfectly correlated with itself), and the off-diagonal elements represent the pairwise correlations between the variables.

```
# Create a matrix of p-values for correlations
cor_pmat <- function(cor_matrix, n){
  # Calculates the p-values of the correlations in a correlation matrix
  # cor_matrix: the correlation matrix
  # n: the number of observations used to compute the correlation matrix

  # Calculate the degrees of freedom
  df <- n - 2

  # Calculate the t-statistic for each correlation
  t_stat <- cor_matrix * sqrt(df / (1 - cor_matrix^2))

  # Calculate the p-value for each correlation
  p_val <- 2 * pt(abs(t_stat), df = df, lower.tail = FALSE)

  # Replace the diagonal with NAs
  diag(p_val) <- NA

  # Return the p-value matrix
  return(p_val)
}

# Obtain P-values for each correlation
p_values <- cor_pmat(cor_matrix, n = nrow(dys))
p_values
```

	Pegboard1	Pegboard2	Pegboard3	Pegboard4	Pegboard5
Pegboard1	NA	0.0002541599	0.01958078	0.006167226	0.23978011
Pegboard2	0.0002541599	NA	0.03973816	0.263313407	0.18356255
Pegboard3	0.0195807763	0.0397381641	NA	0.043909175	0.12126901
Pegboard4	0.0061672259	0.2633134067	0.04390917	NA	0.01543005
Pegboard5	0.2397801056	0.1835625541	0.12126901	0.015430050	NA

- The above code defines a function **cor_pmat** to calculate the p-values for a given correlation matrix and the sample size. The function takes two arguments: **cor_matrix** - the correlation matrix, and **n** - the number of observations used to compute the correlation matrix.
- The function first calculates the degrees of freedom using the sample size. Then, it calculates the t-statistic for each correlation in the matrix. Using the t-statistic, the function then calculates the two-tailed p-value

for each correlation. Finally, it replaces the diagonal elements with NA values and returns the resulting p-value matrix.

- The **cor_pmat** function is then called with **cor_matrix** being the correlation matrix computed earlier in the code, and **n** being the number of rows in the **dys** dataset. The resulting p-value matrix is assigned to the **p_values** variable.

```
# Adjusted significance level using Bonferroni correction
adjusted_alpha <- 0.05 / choose(ncol(cor_matrix), 2)
cat("Adjusted significance level:", adjusted_alpha, "\n")

# Check which pairwise correlations are statistically significant
significant_pairs <- which(p_values < adjusted_alpha, arr.ind = TRUE)
print(significant_pairs)
```

```
Adjusted significance level: 0.005
```

```
      row col
Pegboard2  2  1
Pegboard1  1  2
```

- The output of the code above will depend on the values in the **cor_matrix** and **p_values** variables. However, assuming that **cor_matrix** contains the correlation matrix of some variables and **p_values** contains the p-values for all pairwise correlations among these variables, the code will calculate an adjusted significance level using the Bonferroni correction and then identify which pairwise correlations are statistically significant at this adjusted level.
- The adjusted significance level is calculated by dividing the desired overall significance level (0.05 in this case) by the number of pairwise correlations being tested, which is given by the choose function applied to the number of variables (**ncol(cor_matrix)**) and 2. The resulting adjusted alpha value represents the threshold for significance after taking multiple comparisons into account.
- The **which()** function is then used to identify the indices of the pairwise correlations in **p_values** that are smaller than the adjusted alpha value. These indices correspond to the significant pairwise correlations, and they are printed to the console.

```
# Confidence intervals
# Obtain confidence intervals for each pairwise correlation
for (i in 1:(ncol(cor_matrix)-1)){
  for (j in (i+1):ncol(cor_matrix)){
    cor_test <- cor.test(dys[, i+6], dys[, j+6], method = "pearson", conf.level = 0.95)
    cat("Confidence interval for correlation between Pegboard", i, "and Pegboard", j, ":",
    cor_test$conf.int, "\n")
  }
}
```

- This code computes confidence intervals for pairwise correlations between columns 7 to 12 of the dys data frame, which presumably contains measurements related to dyslexia.
- The for loop iterates over the indices of each pair of columns, and for each pair, it performs a Pearson correlation test using the **cor.test()** function. The **conf.level** argument is set to 0.95, which specifies a 95% confidence interval.
- Note that the dys data frame is assumed to have at least 12 columns, and the variables of interest are assumed to be in columns 7 to 12 (hence the i+6 and j+6 indices in the **cor.test()** function call).

```
# Perform multiple testing correction using the Bonferroni correction
p_adjusted <- p.adjust(p_values, method = "bonferroni")

# Perform multiple testing correction using the FDR correction
p_adjusted <- p.adjust(p_values, method = "fdr")

# Perform multiple testing correction using the Benjamini-Hochberg procedure
```

```
p_adjusted <- p.adjust(p_values, method = "BH")
```

- To perform multiple testing, the Bonferroni correction or the false discovery rate (FDR) correction can be used.
- The Bonferroni correction is a simple approach where you divide the significance level by the number of tests. For example, if you are performing 10 tests with a desired significance level of 0.05, you would use a corrected significance level of $0.05/10 = 0.005$ for each individual test.
- The FDR correction is a more flexible approach that considers the correlation between tests. It controls the expected proportion of false discoveries among all significant results, rather than controlling the type I error rate for each individual test.

The most commonly used method for FDR correction is the Benjamini-Hochberg procedure.

Blog

<Optional: Copy-and-paste any optional Phase 4 Blog posts here>

Chapter 9 – Bayesian Statistics & Markov chains

1 Introduction

Bayesian statistics is a framework for statistical inference that uses probability theory to represent uncertainty and update beliefs as new data is collected. In contrast to classical or frequentist statistics, which relies on fixed parameters and hypotheses, Bayesian statistics allows for the incorporation of prior knowledge, and updating of beliefs based on observed data.

<https://www.youtube.com/watch?v=HZGCoVF3YvM&pp=ygUFQmF5ZXM%3D>

<https://www.youtube.com/playlist?list=PLFDbGp5YzjqXQ4oE4w9GVWdiokWB9gEpm>

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

- $P(A)$ is the prior (distribution); the known information about A before observation of B.
- $P(B|A)$ is the likelihood; the conditional probability of observing the data given the truth.
- $P(B)$ is sometimes referred to as the evidence.
- $P(A)$ and $P(B)$ are known as marginal probabilities ; the probability of observing A and B independently of each other.
- $P(A|B)$ is the posterior probability; the probability of A given that B has been observed or the updated belief in A after taking into account the evidence provided by B.

$$P(A|B) \propto P(A)P(B|A)$$

The posterior is proportional to the prior multiplied by the likelihood.

- In P1 4.1 computes the posterior distribution of a parameter given a prior distribution and some data likelihood.

```
likelihood <- dnorm(x = , mean= , sd= )
```

```
# computes the value of the normal probability density function (PDF) at x = 26.9, with mean mu and standard deviation
```

```
product <- prior * likelihood
```

```
# This multiplies the prior probability distribution by the likelihood to obtain the unnormalized posterior distribution. This is the numerator of Bayes' theorem.
```

```
posterior <- product / sum(product)
```

```
# This normalizes the unnormalized posterior distribution by dividing it by the sum of its values, so that it integrates to 1. This is the denominator of Bayes' theorem.
```

Scenario with data (my MCQ):

The incidence rate of liver cancer among residents in a certain area is 0.0004, and the Alpha-Fetoprotein (AFP) Test is used for the general survey. Medical research shows that there is a

possibility of false detection in laboratory tests. 99% of people with known liver cancer tested positive, while 99.9% of people without liver cancer tested negative. What is the probability of having liver cancer if the first test is positive, and what is the probability of having liver cancer if the two tests are positive?

- Let A ={the patient is a liver cancer patient}, B ={the patient tested positive}, NA ={the patient is NOT a liver cancer patient},
- $P(A)$ =0.0004 (given in the scenario)
- $P(NA)$ =0.9996 (given in the scenario)
- $P(B|A)$ represents the probability of a positive test under the premise of being a liver cancer patient. We already know that is 0.99 from scenario.
- $P(B|NA)$ represents the probability of testing positive under the premise of not being a patient with liver cancer. We already know that is 0.001 from scenario.
- According to the calculation of Bayesian formula, it can be known that:

$$\begin{aligned}P(A|B) &= \frac{P(AB)}{P(B)} = \frac{P(A)P(B|A)}{P(A)P(B|A) + P(NA)P(B|NA)} \\&= \frac{0.0004 \times 0.99}{0.0004 \times 0.99 + 0.9996 \times 0.001} \\&= \frac{0.000396}{0.000396 + 0.0009996} \\&= \frac{0.000396}{0.0013956} \\&= 0.2837\end{aligned}$$

If the re-examination result is still positive, then $P(A)$ =0.2837, $P(NA)$ =0.7163, $P(B|A)$ and $P(B|NA)$ are held constant, brought into the Bayesian formula again.

$$P(A|B)=0.9975$$

Therefore, option B is the correct one, which is why re-examination is necessary in real life.

2. Two forms for the denominator of Bayes' rule: discrete and continuous

<https://www.youtube.com/watch?v=QEzeLh6L9Tg>

In the binomial distribution, the discrete and continuous models differ in the nature of the data being modeled.

In the discrete model, the random variable X represents the number of successes in a fixed number of independent trials, each with the same probability of success. The values of X can only be integer values between 0 and n , where n is the number of trials. The probability mass function (PMF) of the binomial distribution describes the probability of observing each possible value of X .

In the continuous model, the random variable Y represents the proportion of successes in a large number of independent trials. The values of Y can take on any value between 0 and 1, since it represents a proportion. The probability density function (PDF) of the continuous binomial distribution describes the probability density of observing each possible value of Y .

The main difference between the two models is that the discrete model deals with a finite number of possible outcomes (the number of successes) and has a discrete probability distribution, while the continuous model deals with a continuous range of possible outcomes (the proportion of successes) and has a continuous probability distribution. Additionally, the continuous model assumes that the number of trials is very large, such that the probability of observing any specific value of Y is infinitesimally small.

Discrete models \rightarrow sum

continuous models \rightarrow \int

3 Beta distribution in Bayesian inference

In Bayesian inference, the beta distribution is the conjugate prior probability distribution for the Bernoulli, binomial, negative binomial and geometric distributions.

The Beta distribution is a continuous probability distribution, it is defined on the interval $[0,1]$ and has two shape parameters, alpha and beta. The values of these shape parameters determine the shape of the Beta distribution.

The Beta distribution is often used as a prior distribution for a probability parameter, p , in a Bernoulli or Binomial model. The Beta distribution is a conjugate prior to these models, meaning that the posterior distribution is also a Beta distribution. This makes computation of the posterior distribution easier and faster.

The Beta distribution has several important properties, including symmetry, unimodality, and the fact that it can be skewed to the left or right.

4. R codes about Beta distribution in [{LearnBayes}](#) package:

The `dbeta()` function in R is used to evaluate the probability density function (PDF) of the beta distribution at a given point or set of points. The beta distribution is a continuous probability distribution that takes values between 0 and 1 and is often used to model the distribution of probabilities or proportions. The `dbeta()` function takes four arguments:

1. `x`: a numeric vector of values at which to evaluate the PDF.
 2. `shape1`: a numeric value representing the first shape parameter of the beta distribution.
 3. `shape2`: a numeric value representing the second shape parameter of the beta distribution.
 4. `log`: a logical value indicating whether to return the logarithm of the PDF. If TRUE, the function returns the log-density; if FALSE (default), it returns the density.
- Note that the `dbeta()` function only evaluates the PDF at a given point or set of points, the `curve()` function in R to plot the probability density function (PDF) of the Beta distribution with different shape parameters. In addition, it does not calculate probabilities or cumulative probabilities. For those calculations, you can use the `pbeta()` and `qbeta()` functions, respectively.

The `pbeta()` function is used to compute the cumulative distribution function (CDF) of the beta distribution. It takes three arguments:

1. `q` - the quantile(s) at which to evaluate the CDF (i.e., the values for which to compute the probability)
2. `shape1` and `shape2` - the shape parameters of the beta distribution

The `qbeta()` function is used to compute the quantiles of the beta distribution. It takes two arguments:

1. `p` - the credible interval or probabilities for which to compute the quantile(s)
2. `shape1` and `shape2` - the shape parameters of the beta distribution

The `beta.select()` function uses the two percentiles [e.g., ($p=0.5$, $x=0.3$) this percentile is the 50th percentile or the median of the curve, which we believe to be 0.3] to find the shape parameters of the Beta distribution that reflect our beliefs. The resulting shape is a numeric vector with two values, which are the `shape1` and `shape2` parameters of the Beta distribution.

The `rbeta(n, shape1, shape2)` function in R can be used to generate random numbers from a beta distribution.

1. `n` is random numbers generated from a beta distribution with parameters
2. `shape1` and `shape2`. The shape parameters control the shape of the beta distribution. Specifically, `shape1` controls the number of successes and `shape2` controls the number of failures in the prior distribution.

```
sim_2 <- rbeta(100000, shape1 = post_shape[1], shape2=post_shape[2] )
```

```
(sum(sim_2 < 0.5 ) / 100000 ) - pbeta(0.5, shape1 = post_shape[1], shape2=post_shape[2] )
```

estimate the error between using the simulated values in `sim` and the exact answer from the theoretical posterior distribution for probability $p < 0.5$.

the Simulated P1 and P2 from Uniform Priors can be shown through a curve plot using the `dbeta()` function, which plots the density of the beta distribution with specified shape parameters. Also, can be shown through a scatter plot created using the `ggplot()` function from `ggplot2` package(see P3 3.2).

C9_P1 Activity B6

```
set.seed(seed)
sim <- rbeta(10000, shape1 = post_shape[1], shape2 = post_shape[2] )

1 - pbeta(0.75, shape1 = post_shape[1], shape2 = post_shape[2] )
sum(sim > 0.75 ) / 10000

qbeta(c(0.05, 0.95 ), shape1 = post_shape[1], shape2 = post_shape[2] )
quantile(sim, c(0.05, 0.95) )
```

```
qbeta(c(0.025, 0.975 ), shape1 = post_shape[1], shape2 = post_shape[2] )
quantile(sim, c(0.025, 0.975) )

(sum(sim < 0.5 ) / 10000 ) - pbeta(0.5, shape1 = post_shape[1], shape2=post_shape[2] )
(sum(sim < 0.5 ) / 10000 )
pbeta(0.5, shape1 = post_shape[1], shape2=post_shape[2] )
set.seed(seed)
sim_2 <- rbeta(100000, shape1 = post_shape[1], shape2=post_shape[2] )
(sum(sim_2 < 0.5 ) / 100000 )
(sum(sim_2 < 0.5 ) / 100000 ) - pbeta(0.5, shape1 = post_shape[1], shape2=post_shape[2] )
```

1. In this step, we generate 10,000 random draws from the beta distribution with shape parameters `post_shape[1]` and `post_shape[2]` using the function `rbeta()`. The `set.seed()` function is used to ensure that the random numbers generated are reproducible. The resulting values are stored in the variable `sim`.
2. In this step, we calculate the probability that $p > 0.75$ using the simulated values in `sim`. We do this by counting the number of simulated values that are greater than 0.75 and dividing by the total number of simulated values. The result is 0, which means that none of the simulated values were greater than 0.75.
3. In this step, we estimate the error between using the simulated values in `sim` and the exact answer from the theoretical posterior distribution for probability $p < 0.5$. We do this by calculating the difference between the proportion of simulated values that are less than 0.5 and the value obtained using `pbeta()` function. The result is -0.00350916280640168, which means that the proportion of simulated values that are less than 0.5 is slightly smaller than the value obtained from `pbeta()`.
4. In this step, we repeat step 1 but generate 100,000 simulated values instead of 10,000. The resulting values are stored in the variable `sim_2`.
5. In this step, we repeat step 3 using `sim_2` instead of `sim`. The error is now smaller at -0.000749162806401693. This shows that increasing the number of simulated values can lead to more accurate estimates of the posterior distribution.

5. Summary

- Bayes' theorem provides a way to calculate the probability of a hypothesis given some data, by combining our prior beliefs about the hypothesis with the likelihood of the data given the hypothesis.
- The prior distribution represents our beliefs about the parameter before seeing any data. The posterior distribution represents our updated beliefs after seeing the data.
- The choice of prior distribution can have a significant impact on the posterior distribution, and different priors can lead to different conclusions.

*The prior can incorporate expert knowledge or assumptions about the data, but it can also introduce bias if it is not chosen carefully. One way to address the sensitivity of the prior is through sensitivity analysis. Sensitivity analysis can help to identify which aspects of the prior have the greatest impact on the posterior and can inform the choice of a more appropriate prior.

- The posterior distribution can be used to calculate point estimates (such as the mean or median) and credible intervals (similar to confidence intervals in frequentist statistics).
- Markov Chain Monte Carlo (MCMC) methods provide a way to simulate samples from the posterior distribution, allowing for more flexible and complex models.

Blog

In Chapter 9, I learned about Bayesian statistics, a statistical inference framework that provides a way to update our beliefs about parameters of interest based on new data. Bayesian statistics are based on Bayes' theorem, which states that the probability of a hypothesis given some data is proportional to the probability of the data given the hypothesis multiplied by the prior probability of the hypothesis.

One of the main differences between Bayesian statistics and traditional frequentist statistics is that Bayesian statistics explicitly incorporate prior knowledge and beliefs into the analysis, whereas frequentist statistics typically only consider the data at hand. This allows Bayesian statistics to handle situations with limited data, and also allows for a more intuitive interpretation of the results.

I learned to use the Beta prior distribution because the Beta distribution is the conjugate prior to the Bernoulli likelihood function. The posterior is summarised using percentiles and probabilities, based on observing the error between the simulated data and the exact answer from the theoretical posterior distribution, knowing that increasing the number of simulated values leads to a more accurate estimate of the posterior distribution.

I additionally compared two different ways of visualizing the simulated posterior distribution: line plots and scatterplots. The graph shows the posterior distribution for the two coins as a probability density function, while the scatterplot displays the simulated values of the posterior distribution as points in a two-dimensional space.

Course Reflection

Throughout this course, I have achieved the four learning outcomes by gaining a strong understanding of statistical epidemiology and computing. The evidence from each chapter's reflections and blogs recorded how I have achieved.

I have gained a clear understanding of various concepts related to epidemiology, e.g., the epidemiological triad, confounding, randomisation, bias, et al. and apply them in statistical analysis. Through this course, I have successfully achieved this outcome by gaining a comprehensive understanding of these concepts and applying them to real-world datasets.

I have developed my statistical computing skills by learning about mean, standard deviation, variance, covariance, correlation, linear transformations, and probability distribution functions. I have also gained an understanding of Kolmogorov's three axioms of probability, the law of large numbers, and the central limit theorem. After that, I have gained knowledge about maximum likelihood estimators and frequentist and Bayesian statistics. This knowledge has enabled me to understand the importance of these statistical methods in public health research. Like why we especially need confirmatory tests, especially for diseases that have relatively low prevalence.

Although I have some experience in using R, the learning of R language has benefited me a lot, especially the various statistical concepts and functions in R. These included probability distributions such as the Poisson and Binomial distributions, as well as functions for calculating probabilities and likelihoods, such as `dbinom()` and `dnorm()`. We also discussed maximum likelihood estimation using the `mle()` function, and hypothesis testing using functions such as `t.test()` and `chisq.test()`. One common theme throughout these discussions was the importance of understanding and working with probability distributions. Probability distributions provide a way to model the uncertainty inherent in many statistical problems and allow us to make probabilistic statements about the likelihood of certain events or outcomes.

Another key concept I explored a lot was maximum likelihood estimation and frequentist and Bayesian statistics, which are the methods for estimating the parameters of a probability distribution based on observed data. We have learned how to use the `mle()` function in R to perform maximum likelihood estimation, and how to define a log-likelihood function to optimize the parameters as well as how to apply Bayes rule to obtain posterior densities by combining prior belief with data or summarise posterior distributions using percentiles, probabilities, and simulations..

Throughout the course I worked through exploratory data analysis example after example, exercising my ability to visualize and analyze data using R. Reviewed and gained a deeper understanding of how to use R functions such as `cor()` and `hist()` to calculate correlations and generate histograms, how to use the `qqnorm()` function to generate Q-Q plots to check for normality, and more.

MCQ discussions and vivid explanations on YouTube are good supplements to the course. In addition, I gained a lot of fun in the process of challenging additional questions and reading extracurricular materials. It is a bit regret that I am not being able to complete the mind map in time because I have a full-time job, but I will continue to finish it.

HDAT9200 Statistical Foundations for Health Data Science

Course Reflection including collation of Phase 4 (chapter reflection and blog) posts

Reference:

Jovel, J., et al. (2016). "Characterization of the Gut Microbiome Using 16S or Shotgun Metagenomics." Front Microbiol **7**(459): 17.