Test evaluation without a gold standard

Eleftherios Meletis

26 October 2023

Recap from Day 1

- Sensitivity, Specificity - definitions
- Bayes theorem - Markov Chain Monte Carlo
- Apparent & true prevalence estimation

Familiar with this JAGS model?

```
ap_model <-
'model {

  # Define likelihood distribution of the data
  # JAGS Binomial distribution Arguments: ap, n

  y ~ dbin(ap,n)

  # Specify prior distribution for parameters of interest
  # Uniform (non-informative) prior distribution
  ap ~ dbeta(1,1)

  #data# n, y
  #monitor# ap
  #inits# ap
}
'
```

What about this one?

```
tp_model <-
'model {

  # Define likelihood distribution of the data
  # JAGS Binomial distribution Arguments: ap, n
  y ~ dbin(ap,n)

  ap <- tp * Se + (1-tp)*(1-Sp)
  # Specify prior distribution for parameters of interest
  # Uniform (non-informative) prior distribution
  tp ~ dbeta(1,1)

  # Prior distributions for Se, Sp
  Se ~ dbeta(100,9)
  Sp ~ dbeta(100,9)

  #data# n, y
  #monitor# tp, Se, Sp
  #inits# tp, Se, Sp
}
'
```

- Do you see a something weird here (degrees of freedom vs parameters of interest)
- Add a second test and a second population
- Key assumptions of a latent class model

Sensitivity - Specificity estimation with and without a gold standard

Hui-Walter paradigm/model (1980)

- A particular model formulation that was originally designed for evaluating diagnostic tests in the absence of a gold standard
- Not originally/necessarily Bayesian - implemented using Maximum Likelihood
- If we don't know the true disease status, how can we estimate sensitivity or specificity for either test?

Hui-Walter paradigm (1980)

- But we will use the data/observations from the manuscript published back in 1980.

Hui-Walter (1980) dataset

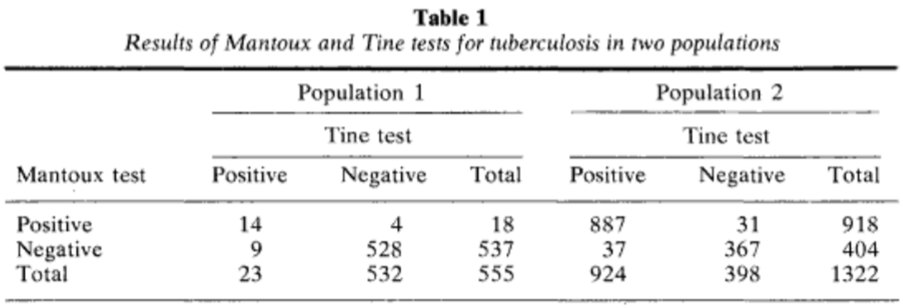


Table 1
Results of Mantoux and Tine tests for tuberculosis in two populations

	Population 1			Population 2		
	Tine test			Tine test		
Mantoux test	Positive	Negative	Total	Positive	Negative	Total
Positive	14	4	18	887	31	918
Negative	9	528	537	37	367	404
Total	23	532	555	924	398	1322

Encode the Table_1 data in RStudio

```
pop_1 = matrix(nrow=3,ncol=3)
rownames(pop_1) = c("Mantoux_Test_Pos", "Mantoux_Test_Neg", "Total")
colnames(pop_1) = c("Tine_Test_Pos", "Tine_Test_Neg", "Total")

pop_1[1,1] = 14
pop_1[1,2] = 4
pop_1[2,1] = 9
pop_1[2,2] = 528
#Total rows and columns
pop_1[1,3] = pop_1[1,1] + pop_1[1,2]
pop_1[2,3] = pop_1[2,1] + pop_1[2,2]
pop_1[3,1] = pop_1[1,1] + pop_1[2,1]
pop_1[3,2] = pop_1[1,2] + pop_1[2,2]
N_1 = sum(pop_1[1,1] + pop_1[1,2] + pop_1[2,1] + pop_1[2,2])
pop_1[3,3] = N_1
pop_1
```

```
##                  Tine_Test_Pos Tine_Test_Neg Total
## Mantoux_Test_Pos            14             4    18
## Mantoux_Test_Neg             9           528   537
## Total                       23           532   555
```

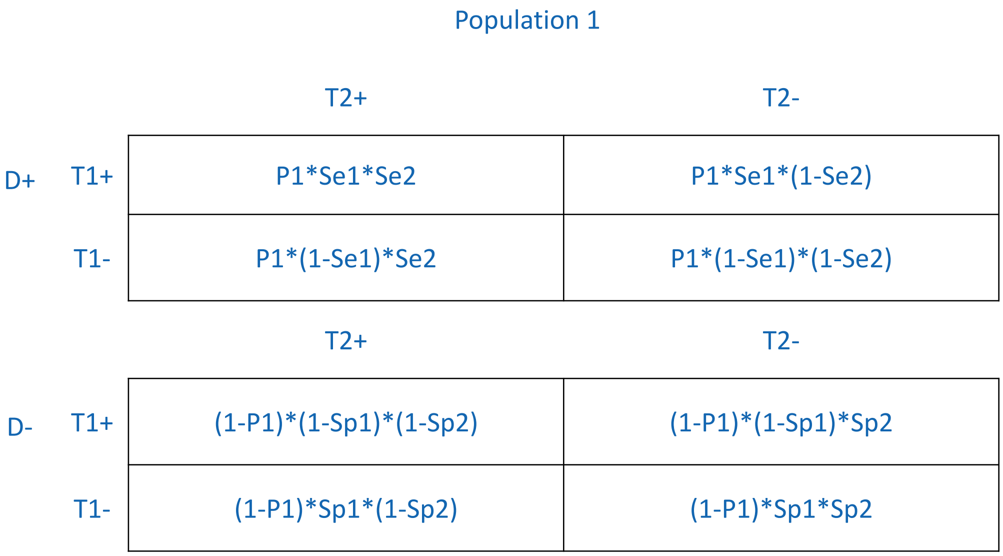
Now let's do pop_2

```
pop_2 = matrix(nrow=3,ncol=3)
rownames(pop_2) = c("Mantoux_Test_Pos", "Mantoux_Test_Neg", "Total")
colnames(pop_2) = c("Tine_Test_Pos", "Tine_Test_Neg", "Total")

pop_2[1,1] = 887
pop_2[1,2] = 31
pop_2[2,1] = 37
```

```
pop_2[2,2] = 367
#Total rows and columns
pop_2[1,3] = pop_2[1,1] + pop_2[1,2]
pop_2[2,3] = pop_2[2,1] + pop_2[2,2]
pop_2[3,1] = pop_2[1,1] + pop_2[2,1]
pop_2[3,2] = pop_2[1,2] + pop_2[2,2]
N_2 = sum(pop_2[1,1] + pop_2[1,2] + pop_2[2,1] + pop_2[2,2])
pop_2[3,3] = N_2
pop_2
```

```
##                   Tine_Test_Pos Tine_Test_Neg Total
## Mantoux_Test_Pos            887            31   918
## Mantoux_Test_Neg             37           367   404
## Total                       924           398  1322
```

Hui-Walter paradigm (1980)



Population 1

		T2+	T2-
D+	T1+	P1*Se1*Se2	P1*Se1*(1-Se2)
	T1-	P1*(1-Se1)*Se2	P1*(1-Se1)*(1-Se2)
		T2+	T2-
D-	T1+	(1-P1)*(1-Sp1)*(1-Sp2)	(1-P1)*(1-Sp1)*Sp2
	T1-	(1-P1)*Sp1*(1-Sp2)	(1-P1)*Sp1*Sp2

Hui-Walter paradigm (1980)

Population 1

T1+T2+:	P1*Se1*Se2+(1-P1)*(1-Sp1)*(1-Sp2)
T1+T2-:	P1*Se1*(1-Se2)+(1-P1)*(1-Sp1)*Sp2
T1-T2+:	P1*(1-Se1)*Se2+(1-P1)*Sp1*(1-Sp2)
T1-T2-:	P1*(1-Se1)*(1-Se2)+(1-P1)*Sp1*Sp2

- 5 parameter and 3 degrees of freedom
  - Non identifiable model

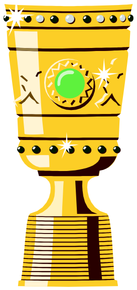
Hui-Walter paradigm (1980)

Population 1

T1+T2+:	P1*Se1*Se2+(1-P1)*(1-Sp1)*(1-Sp2)
T1+T2-:	P1*Se1*(1-Se2)+(1-P1)*(1-Sp1)*Sp2
T1-T2+:	P1*(1-Se1)*Se2+(1-P1)*Sp1*(1-Sp2)
T1-T2-:	P1*(1-Se1)*(1-Se2)+(1-P1)*Sp1*Sp2

Population 2

T1+T2+:	P2*Se1*Se2+(1-P2)*(1-Sp1)*(1-Sp2)
T1+T2-:	P2*Se1*(1-Se2)+(1-P2)*(1-Sp1)*Sp2
T1-T2+:	P2*(1-Se1)*Se2+(1-P2)*Sp1*(1-Sp2)
T1-T2-:	P2*(1-Se1)*(1-Se2)+(1-P2)*Sp1*Sp2

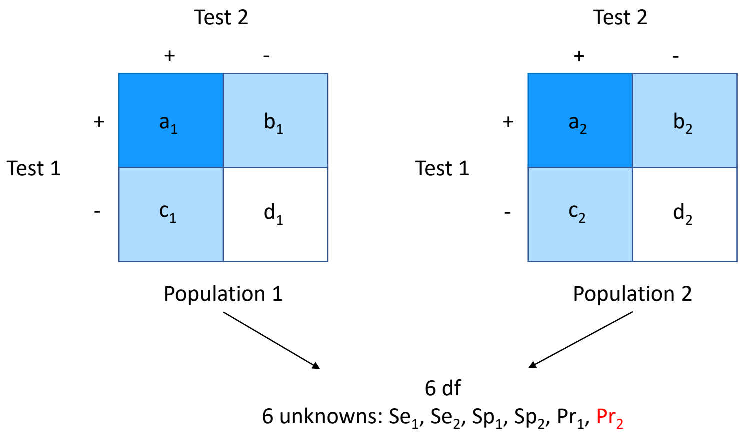


Identifiable model!

6=6

Hui-Walter model

- A particular model formulation that was originally designed for evaluating diagnostic tests in the absence of a gold standard
- Also known as the two_test - two_population setting/paradigm



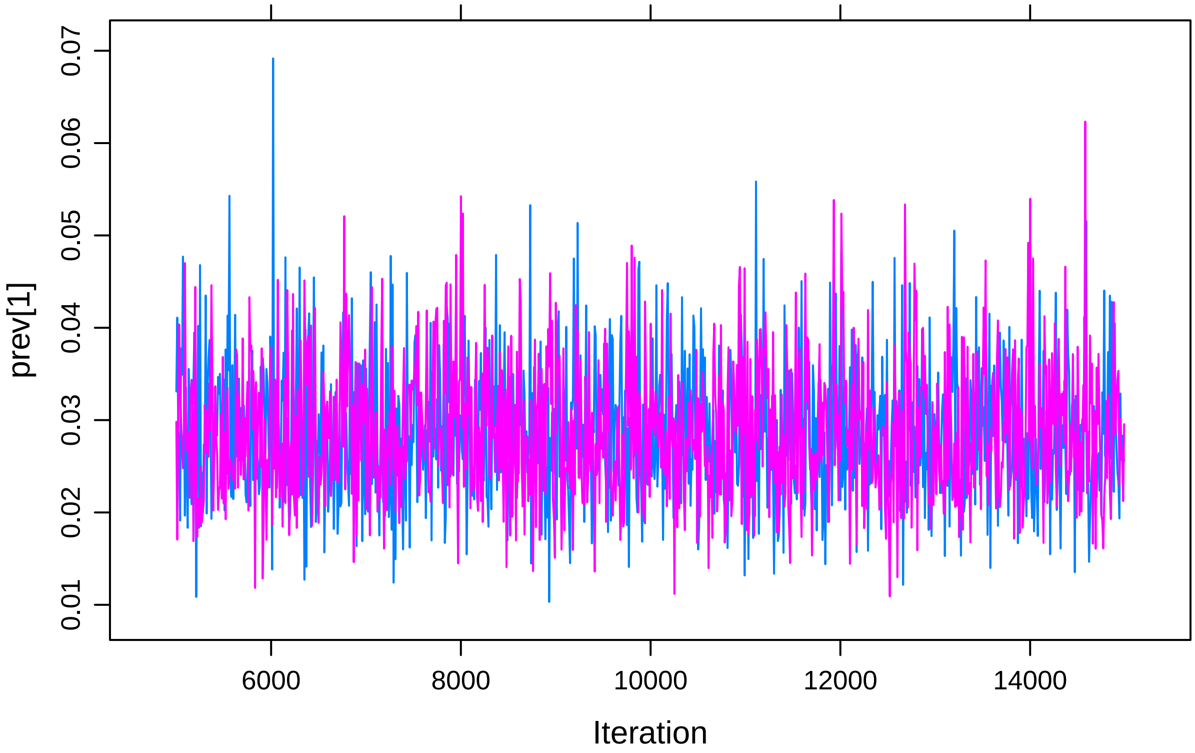
Model Specification ('hw_definition')

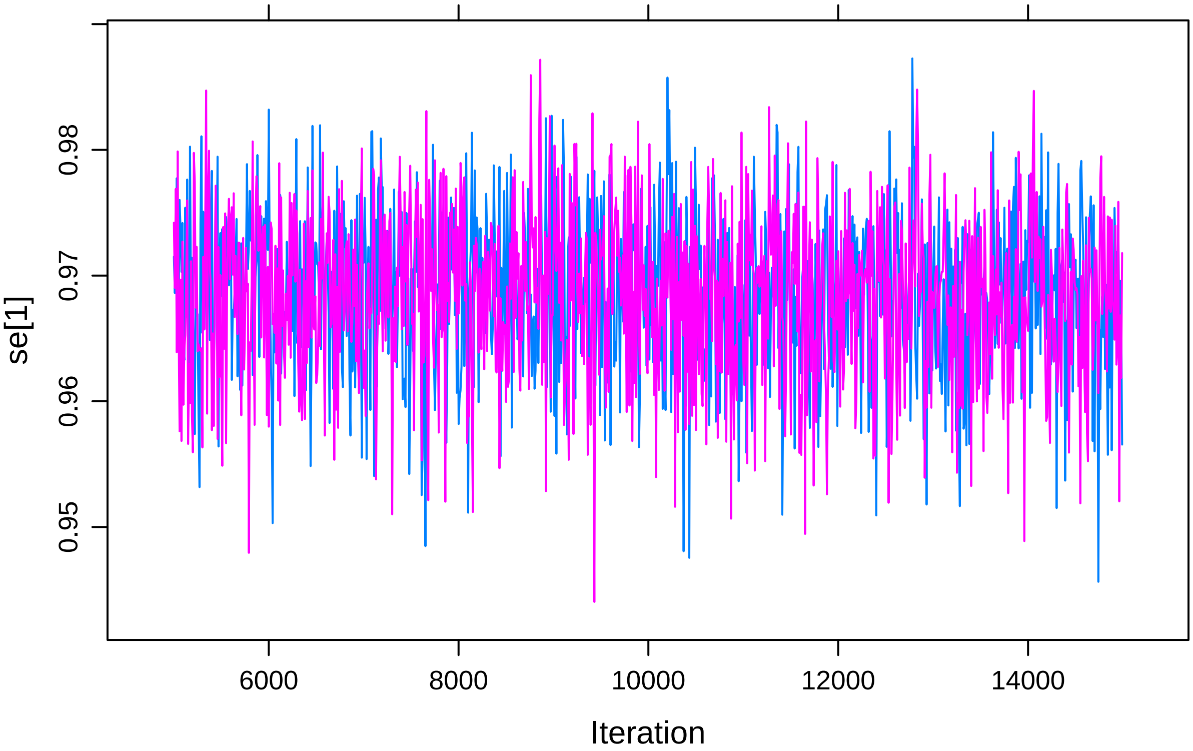
```
hw_definition <- c("model{
  Population_1 ~ dmulti(prob_1, N_1)
  Population_2 ~ dmulti(prob_2, N_2)

  #Population_1

  # Test1+ Test2+
    prob_1[1] <- (prev[1] * ((se[1])*(se[2]))) + ((1-prev[1]) * ((1-sp[1])*(1-sp[2])))

  # Test1+ Test2-
    prob_1[2] <- (prev[1] * ((se[1])*(1-se[2]))) + ((1-prev[1]) * ((1-sp[1])*(sp[2])))

  # Test1- Test2+
    prob_1[3] <- (prev[1] * ((1-se[1])*(se[2]))) + ((1-prev[1]) * ((sp[1])*(1-sp[2])))

  # Test1- Test2-
    prob_1[4] <- (prev[1] * ((1-se[1])*(1-se[2]))) + ((1-prev[1]) * ((sp[1])*(sp[2])))

    #Population_2
  # Test1+ Test2+
    prob_2[1] <- (prev[2] * ((se[1])*(se[2]))) + ((1-prev[2]) * ((1-sp[1])*(1-sp[2])))

  # Test1+ Test2-
    prob_2[2] <- (prev[2] * ((se[1])*(1-se[2]))) + ((1-prev[2]) * ((1-sp[1])*(sp[2])))

  # Test1- Test2+
    prob_2[3] <- (prev[2] * ((1-se[1])*(se[2]))) + ((1-prev[2]) * ((sp[1])*(1-sp[2])))

  # Test1- Test2-
    prob_2[4] <- (prev[2] * ((1-se[1])*(1-se[2]))) + ((1-prev[2]) * ((sp[1])*(sp[2])))

  prev[1] ~ dbeta(1, 1)
  prev[2] ~ dbeta(1, 1)

  se[1] ~ dbeta(1, 1)T(1-sp[1], )
  sp[1] ~ dbeta(1, 1)
  se[2] ~ dbeta(1, 1)T(1-sp[2], )
  sp[2] ~ dbeta(1, 1)

  #data# Population_1, Population_2, N_1, N_2
  #monitor# prev, prob_1, prob_2, se, sp
  #inits# prev, se, sp
}
")
```

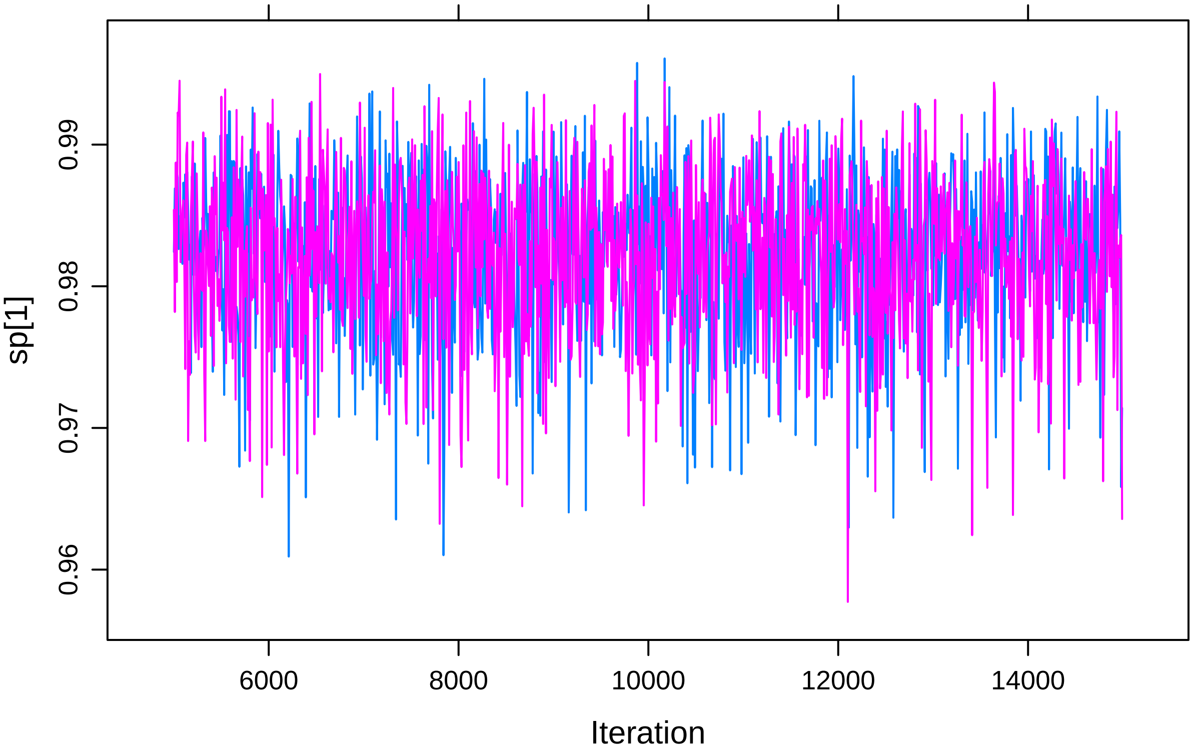
```
library('runjags')

Population_1 <- as.numeric(pop_1[1:2,1:2])
Population_2 <- as.numeric(pop_2[1:2,1:2])


prev <- list(chain1=c(0.05,0.99), chain2=c(0.95,0.05))
se <- list(chain1=c(0.5,0.99), chain2=c(0.99,0.5))
sp <- list(chain1=c(0.5,0.99), chain2=c(0.99,0.5))

results <- run.jags(hw_definition, n.chains=2)
```

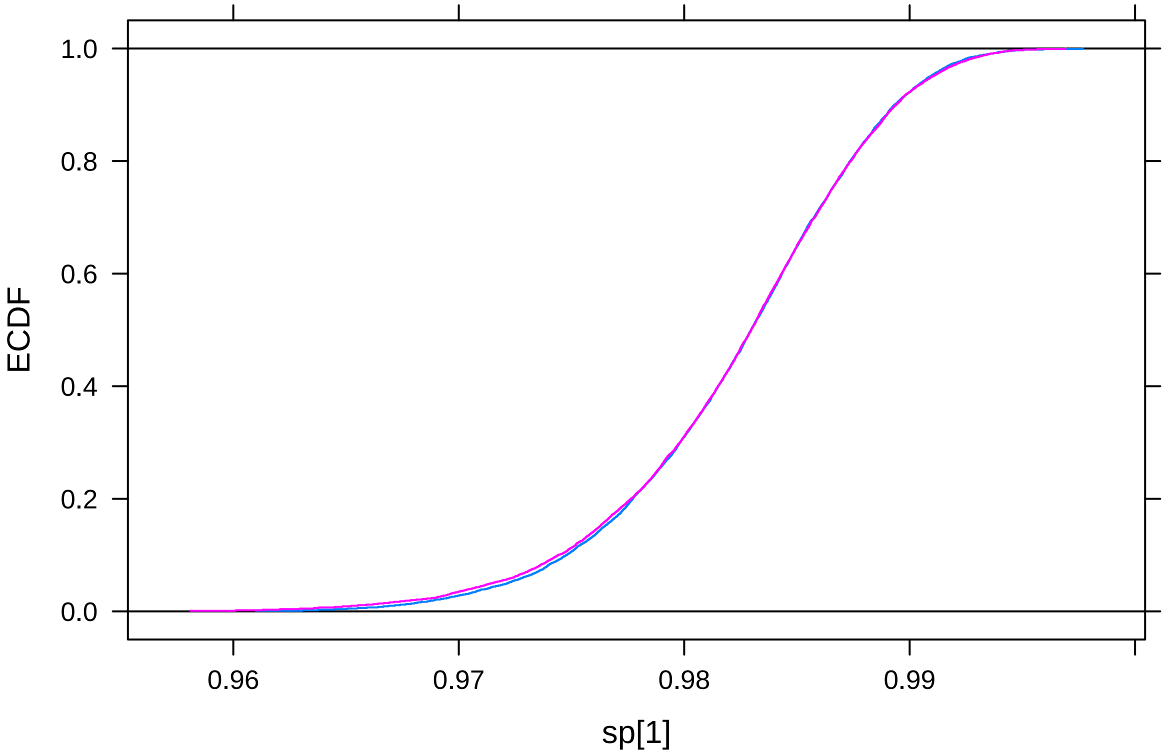
Remember to check convergence and effective sample size!

```
plot(results)
```


prev[1]
0.01
0.02
0.03
0.04
0.05
0.06
0.07
6000
8000
10000
12000
14000
Iteration


se[1]
0.95
0.96
0.97
0.98
6000
8000
10000
12000
14000
Iteration


0.96
0.97
0.98
0.99
sp[1]
6000
8000
10000
12000
14000
Iteration



```
summary(results)
```

Exercise

Run the model and compare the results we the ones obtained from the original Hui-Walter model.

Points to discuss

1. What is the latent variable here - also discuss about it tomorrow.
2. Can this type of models support more tests and more populations?
3. What is conditional (in)depedence between diagnostic tests and can we adjust for that?
4. Three main assumptions of this model

Any questions?

Video Summary - Take home message

https://www.youtube.com/watch?v=z6devQmW2xE&ab_channel=PolychronisKostoulas