# Bayesian Estimation of Vaccine Efficacy Using JAGS

Emerson Rodrigues Vero filho, Roya Joulaei Vijouyeh

University of the Studies of Padova

September 25, 2023

# Task

Several Covid-19 Vaccines have been authorized by the European Medicines Agency (EMA):

- Nuvaxovid (Novavax CZ) [1];
- Vaxzevria (AstraZeneca AB) [2];
- Spikevax (Moderna Biotech) [3]
- Comirnaty (BioNTech/Pfizer) [4]
- Jcovden (Janssen-Cilag) [5]

Similar information is provided in the United States Food and Drugs web site [6]. For the project:

1. Collect official data available on the clinical clinical trial performed for each vaccine and compute with JAGS or Stan the efficacy of each Vaccine. Infere the the 95% credibility interval.
2. More recently tests on the efficacy of Vaccine for young people have started. Try to collect available official data from the European medicines Agency (https://www.ema.europa.eu/en) or the U.S. Food and Drug (FDA) (https://www.fda.gov/) and perform a Bayesian analysis of the data as a function of the age of the patients.

# Markov Chain Monte Carlo (MCMC) Approach

To solve the problem, a Markov Chain Monte Carlo (MCMC) approach was used. More specifically, a simulation model was built with R's JAGS framework (Just Another Gibbs Sampler).

- MCMC is a powerful method for sampling from complex probability distributions.
- It is used when direct sampling is difficult or impossible.
- The Gibbs sampler is a specific type of MCMC algorithm.
- It generates samples from a multivariate distribution by iteratively sampling each variable from its conditional distribution given the current values of the other variables.

The JAGS framework provides a flexible and efficient tool for implementing Bayesian analysis using the Gibbs sampler.

# Collecting Data

At first we will do all steps for one of the vaccines and then we will do same computatuion for all vaccine which mentioned.

**Table 2: Vaccine efficacy against PCR-confirmed COVID-19 with onset from 7 days after second vaccination [1] - PP-EFF analysis set; Study 2019nCoV-301**

| Subgroup | Nuvaxovid | | | Placebo | | | % Vaccine Efficacy (95% CI) |
|---|---|---|---|---|---|---|---|
| | Partici-pants N | COVID-19 cases n (%)[2] | Incidence Rate Per Year Per 1,000 People[2] | Partici-pants N | COVID-19 cases n (%)[3] | Incidence Rate Per Year Per 1,000 People[2] | |
| **Primary efficacy endpoint** | | | | | | | |
| All participants | 17,312 | 14 (0.1) | 3.26 | 8,140 | 63 (0.8) | 34.01 | 90.4% (82.9, 94.6)[3,4] |

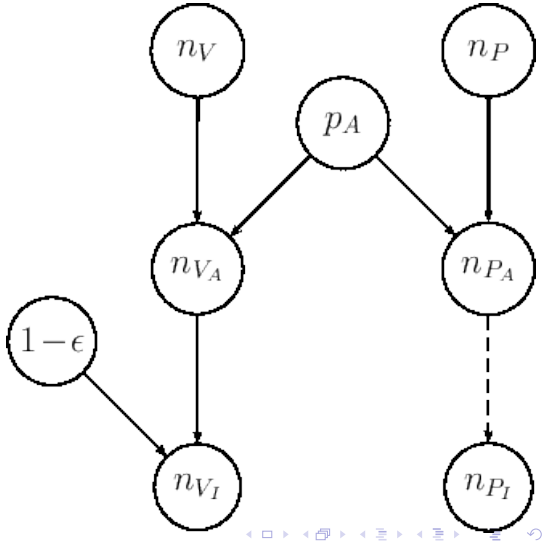
Figure: Data for Nuvaxovid (Novavax CZ) :

A placebo is an inactive substance that looks like the drug or treatment being tested. Comparing results from the two groups suggests whether changes in the test group result from the treatment or occur by chance.

# Model

We begin by drawing a Directed Acyclic Graph (DAG) model that can represent our problem. What is described in red is the data that we have.

- nV: number of people in vaccine group
- nP: number of people in placebo group
- pA: probability to get assaulted by the virus (a catch all term)
- nP.I: number of people in placebo group who got infected
- nV.I: number of people in vaccine group who got infected
- nVa: people who got assaulted by the virus from the vaccinated group.
- epsilon: efficacy of the vaccine = $1 -$ prob



# Model

- Question: How to relate the numbers of infected to the numbers of the participants in the trial?
- Simple Answer: a catch all term pA embedding the many real life variables, apart being vaccinated or not.

Nodes $n_{V_A}$ and $n_{P_A}$ represent them the number of 'assaulted individuals' in each group, and they are modeled according to a binomials distributions, that is:

$$n_{V_A} \sim \text{Binom}(n_V, p_A) \ (1)$$
$$n_{P_A} \sim \text{Binom}(n_P, p_A) \ , \ (2)$$

Instead, the 'assaulted individuals' of the other group are 'shielded' by the vaccine, with probability of being infected equal to $1 - \epsilon$, where $\epsilon$ is the efficacy:

$$n_{V_I} \sim \text{Binom}(n_{V_A}, 1-\epsilon). \ (3)$$

# JAGS for Bayesian Analysis

- JAGS simplifies Bayesian analysis by focusing on model description and transparent instructions.
- Required data for the analysis includes:
- JAGS samples the space of possibilities based on the provided model and data.
- It returns lists of numbers (chains) for each monitored variable.
- The frequency of values in each chain reflects the probability distribution of the variable.
- The model architecture is specified using BUGS language.

```
modelinho = "tmp_model.bug"              #create bug language file and write on it
write("
model {
  nV.A  ~ dbin(pA, nV)     #this amount would have gotten covid if there wasn't the vaccine
  nP.I   ~ dbin(pA, nP)    #this amount also, and there wasn't in fact, because they got placebo
  pA ~ dbeta(0.0101010,1)  #we assume a 1% chance of contamination, with relatively weak prior beliefs        #prior
  nV.I ~ dbin(ffe, nV.A)   #the efficacy of the vaccine filters the people who would have gotten covid
  ffe ~ dbeta(1,1)         #ffe is the shield parameter
  eff <- 1 - ffe           #ffe is defined as 1 minus the efficacy                                           #prior
}
", modelinho)     #write bugs model in the file you have just created
```

Figure: Model Code

# Prior Beliefs

Our prior beliefs assume that the probability of getting COVID-19 with a vaccine and with a placebo are identical. We model these probabilities using the Beta distribution with parameters $\alpha$ and $\beta$.

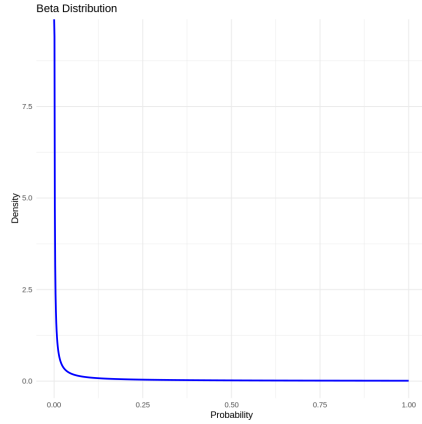
To ensure weak prior beliefs that can be updated with data, we select the parameters such that:

$$\frac{\alpha}{\beta+1} = \frac{\alpha}{\alpha+1} \approx \frac{1}{100}$$

$\alpha \approx 0.010101$.

So, We arrive at a prior of Beta(0.010101, 1) distribution.

We also use a beta distribution Beta(1, 1) for the vaccine efficacy.



# Initializing the Model

```
# Observed Data For the Model
nP   = 21728     # number of people in the placebo group
nV   = 21720     # number of people in the vaccine group
nP.I = 162       # number of people infected in the placebo group
nV.I = 8         # number of people infected in the vaccine group

data <- list(nP=nP, nV=nV, nP.I=nP.I, nV.I=nV.I)
```

```
jm <- jags.model(modelinho, data)     #feeding jags with the model and the data, jags model initializes
update(jm, 100)
```

```
Compiling model graph
   Resolving undeclared variables
   Allocating nodes
Graph information:
   Observed stochastic nodes: 2
   Unobserved stochastic nodes: 3
   Total graph size: 10

Initializing model
```

```
to.monitor <-  c('pA', 'eff', 'nV.A')      #telling it which variables we want to monitor
```

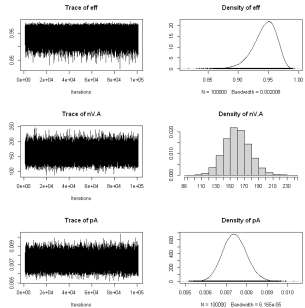
```
chain <- coda.samples(jm, to.monitor, n.iter=nr)   #output is a mcmc object
```

# The Chain Object

```
class(chain)
methods(class = "mcmc.list")
```

```
'mcmc.list'
 [1] [             acfplot       as.array      as.matrix     as.mcmc
 [6] autocorr.diag batchSE       end           hdi           head
[11] HPDinterval   plot          rejectionRate start         summary
[16] tail          thin          time          window
see '?methods' for accessing help and source code
```

```
[[1]]
Markov Chain Monte Carlo (MCMC) output:
Start = 1101
End = 101100
Thinning interval = 1
                eff nV.A          pA
     [1,] 0.9334688  150 0.007432370
     [2,] 0.9326831  182 0.008346539
     [3,] 0.9505243  161 0.008362878
     [4,] 0.9359299  183 0.007653828
     [5,] 0.9402617  164 0.007548849
     [6,] 0.9441624  168 0.007475072
     [7,] 0.8946423  170 0.007225271
     [8,] 0.9361768  150 0.006934321
     [9,] 0.9274185  149 0.007186532
    [10,] 0.9527095  129 0.006570882
    [11,] 0.9517227  136 0.006994959
    [12,] 0.9159106  131 0.006772122
    [13,] 0.9139023  126 0.006484283
    [14,] 0.9560245  156 0.007209435
    [15,] 0.9597561  158 0.006766965
    [16,] 0.9370360  156 0.007782018
    [17,] 0.9523519  172 0.007591247
```



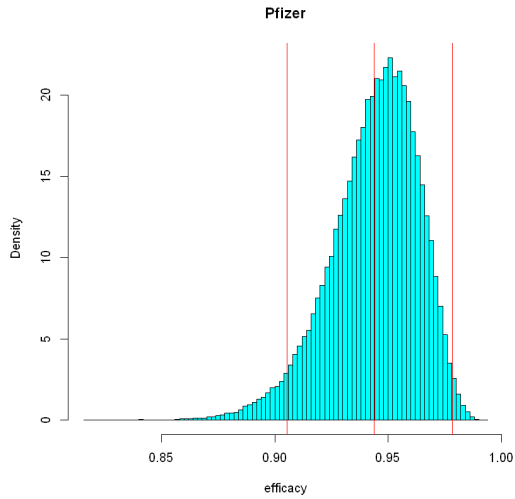

| | eff | nV.A | pA |
|---|---|---|---|
| **Lag 0** | 1.000000000 | 1.000000000 | 1.000000000 |
| **Lag 1** | 0.110021149 | 0.529491825 | 0.623417151 |
| **Lag 5** | 0.017045604 | 0.085521716 | 0.105478118 |
| **Lag 10** | 0.007482657 | 0.011797620 | 0.012238498 |
| **Lag 50** | 0.002284069 | 0.004368132 | 0.007029575 |

# Plotting the Distribution

- $nP = 21728$: Number of people in the placebo group
- $nV = 21720$: Number of people in the vaccine group
- $nP.I = 162$: Number of people infected in the placebo group
- $nV.I = 8$: Number of people infected in the vaccine group

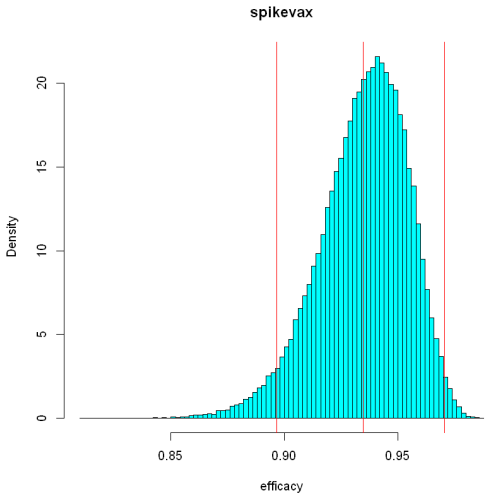
```
chain.df.Pfizer <- as.data.frame( as.mcmc(chain) )
# Extract the efficacy samples from the chain
eff_samples <- chain.df.Pfizer$eff

# Calculate the 95% credibility interval using the HDI function
hdi <- hdi(eff_samples, credMass = 0.95)

# Print the HDI
cat("95% Interval of Confidence: [", hdi[1], ", ", hdi[2], "]\n")

cat(paste("mean: ", mean(chain.df.Pfizer$eff)))

hist(chain.df.Pfizer$eff, nc = 100, col = 'cyan', freq = FALSE,
     xlab = 'efficacy', main = 'Pfizer')

horizontal <- c(hdi[1], mean(chain.df.Pfizer$eff), hdi[2])
abline(v = horizontal, col = "red")
```

```
95% Interval of Confidence: [ 0.9046278 ,  0.9774333 ]
mean:  0.943650306816385
```

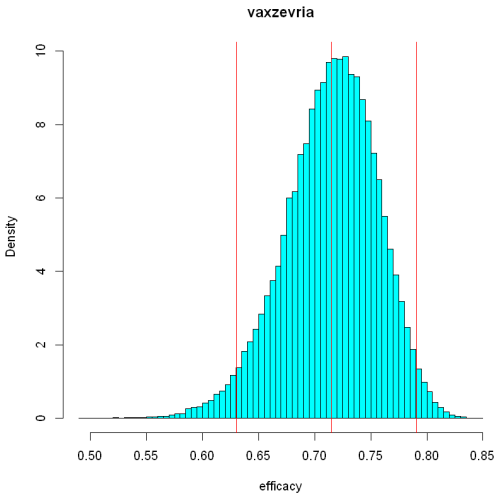


# Other Results for Efficacy (Part 1)

mean: 0.934
95% Interval of Confidence: [0.896 , 0.970]

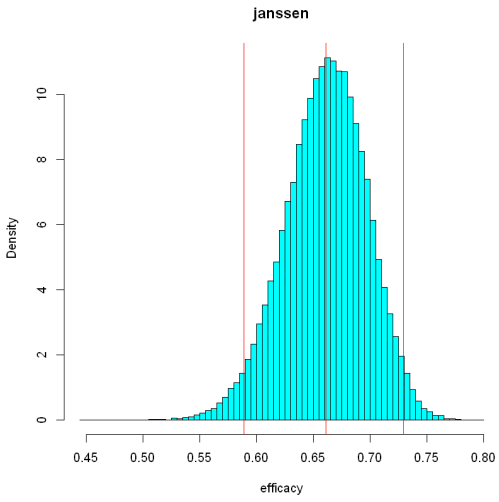



mean: 0.715
95% Interval of Confidence: [ 0.63 , 0.79]

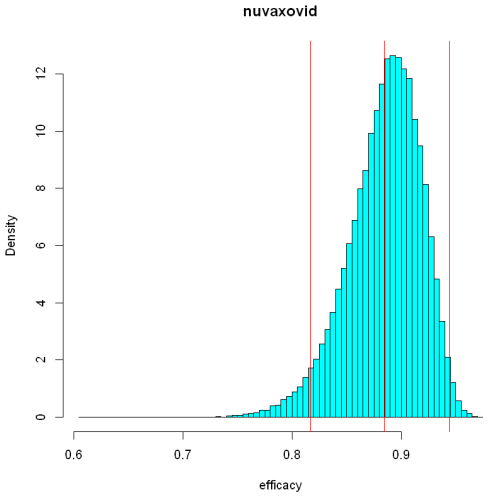


# Other Results for Efficacy (Part 2)

mean: 0.660 95% Interval of Confidence: [0.589 , 0.729]




mean: 0.885
95% Interval of Confidence: [0.817 , 0.944]



# Assault Probability

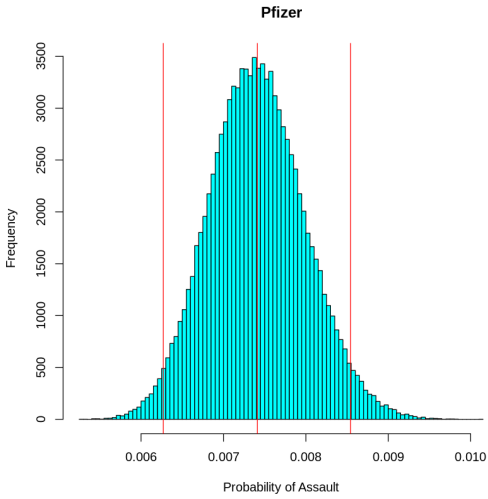



Figure: Assault Probability for Pfizer

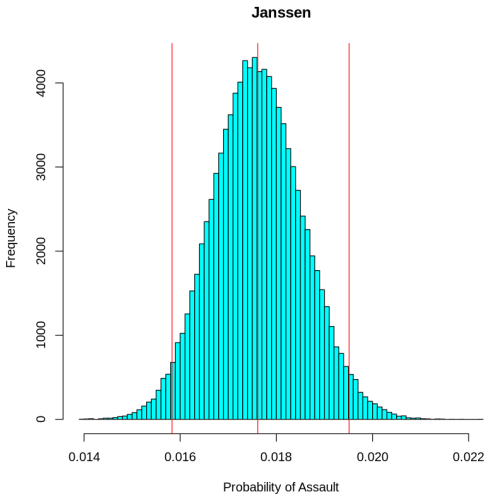



Figure: Assault Probability for Janssen

# Task 2

We perform the same computation using data for different age groups and examine the correlation between age and vaccine efficacy.

```
[ ]  #data for different age groups (pfizer study)
     age_range<-c("<16", "16-55", "55-65", "65-75", ">75")
     nP   <- c(331, 11533, 8208, 4226, 847)
     nV   <- c(342, 11517, 8194, 4192, 842)
     nP.I <- c(10, 568, 266, 124, 26)
     nV.I <- c(0, 52, 25, 7, 1)

     #data<- list(nP=nP[i], nV=nV[i], nP.I=nP.I[i], nV.I=nV.I[i])
     #the data for a certain age_range is given by its index in age_range list
```

```r
per_age_estimator<-function(i){
data<- list(nP=nP[i], nV=nV[i], nP.I=nP.I[i], nV.I=nV.I[i])
jm <- jags.model(modelinho, data)
update(jm, 100)
to.monitor <-  c('pA', 'eff', 'nV.A')
chain <- coda.samples(jm, to.monitor, n.iter=nr)
chain.df.Pfizer <- as.data.frame( as.mcmc(chain) )
eff_samples <- chain.df.Pfizer$eff
mean_eff <- mean(eff_samples)
return (mean_eff)
}



# Estimate vaccine efficacy for each age group
result <- vector()
for (i in seq_along(age_range)) {
  mean_eff <- per_age_estimator(i)
  result <- c(result, mean_eff)
}

# Print the estimated vaccine efficacy for each age group
for (i in seq_along(age_range)) {
  cat("Age Group:", age_range[i], "\tVaccine Efficacy:", round(result[i], 4), "\n")
}
```

```r
vaccine_efficacy <- vector()

for (i in seq_along(age_range)) {
  mean_eff <- per_age_estimator(i)
  vaccine_efficacy <- c(vaccine_efficacy, mean_eff)
}

# Create a barplot of vaccine efficacy by age range
barplot(vaccine_efficacy, names.arg = age_range, xlab = "Age Range",
        ylab = "Vaccine Efficacy", ylim = c(0.85, 0.95),
        main = "Vaccine Efficacy by Age Group")

# Calculate the x-coordinates of the center points
center_points <- barplot(vaccine_efficacy, plot = FALSE)

# Add a line graph representing the vaccine efficacy
lines(center_points, vaccine_efficacy, type = "b", col = "blue", lwd = 2)
```

# Vaccine Efficacy per Age Group

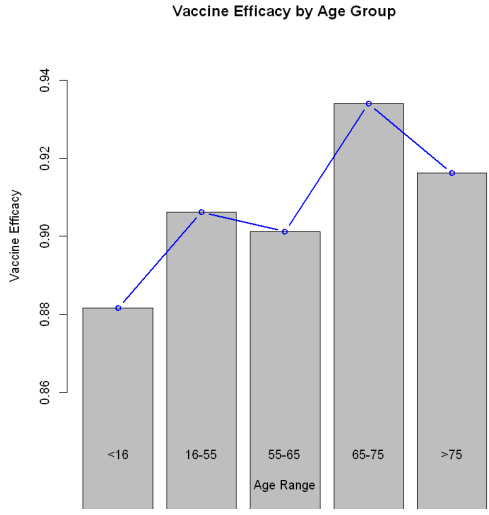
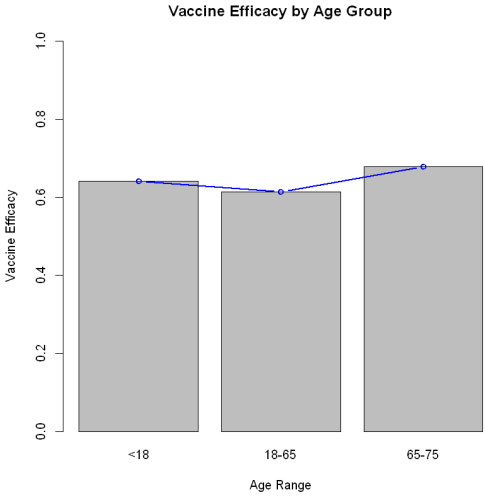
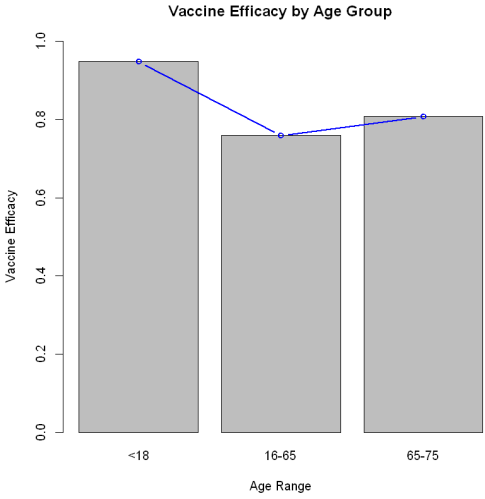



Figure: Pfizer

# Other Results




Janssen




Spykevax

# Conclusions

- Pfizer Shows the best results for Vaccine Efficacy. Janssen shows the worst.
- Our model did not suggest a clear correlation between group age and vaccine efficacy
- We have encountered different assault probabilites for the different data. So the results for vaccine efficacy are influenced by them and are better understood if referenced for those specific populations.