Hib Efficacy

Arminder K Deol

29/04/2019

Hib effiacy

The Hib_efficacy code authored by A Deol on Github was derived from the rotavirus_vaccine_efficacy code authored by K Van Zandvoort. The purpose is to estimate the cumulative vaccine efficacy as well as the instantaneous vaccine effacy - using various methods for the latter. The waning function is changed inside the rotavirus_vaccine_efficacy_index.r. Note that the file names did not change from Kevin's original names - the rotavirus reference is in fact Hib.

A GLMM model was used to for to cumulative vaccine efficacy at different time-frames, using a Bayesian framework.

```
library(data.table)
library(ggplot2)
library(rjags)
library(Rcpp)
library(gridExtra)
```

Average VE is calculated by 1-RR, but this is not time-specific. Here c++ was used to convert cumulative vaccine efficacy (VE) to instantaneous vaccine efficacy (iVE) at a given time point, in the presence of waning.

```
#include <Rcpp.h>
using namespace Rcpp;

// [[Rcpp::export]]

NumericVector cumVEtoInsFast(
    Rcpp::NumericVector ve_c,
    Rcpp::NumericVector rate_p
){
    int ve_c_N = ve_c.size();
    Rcpp::NumericVector ve_i(ve_c_N);

    int rate_N = rate_p.size();
    long double diff = 0.0;
    long double ratediff = 0.0;

    int t = 0;
    int z = 0;

    for( t = 0; t < ve_c_N; t++ ){
        if(t == 0){
            ve_i[t] = ve_c[t];
        } else {
            diff = 0.0;
            for( z = 0; z < t; z++ ){
                if(rate_N == 1){
                    ratediff = 1.0;
                    diff += (ve_c[t] - ve_i[z]);
```

```
                } else {
                    ratediff = rate_p[z]/rate_p[t];
                    diff += (ve_c[t] - ve_i[z])*ratediff;
                }
            }
            ve_i[t] = ve_c[t] + diff;
        }
    }
    return(ve_i);
}
```

This is where the dataframe is read in (rotavirus_vaccine_efficacy_extracted.csv is in face Hib data but with file name unchanged from Kevin's original code) and prepared for analysis and where the RR is calculated.

```
#prepare extracted data pooled analysis
#We re-calculate the RR in exactly the same way in each study period.
#We need to calculate the relative risk, due to the sparse follow-up
# data, and because censored number of people is not consistently
# available in each study.
#Will create a data-table with the cleaned and processed estimates: vaccine_efficacy_data

vaccine_efficacy_data <- fread("rotavirus_vaccine_efficacy_extracted.csv")
vaccine_efficacy_data[,"study_id"] <- 0
vaccine_efficacy_data[,"follow_up_p"] <- 0
vaccine_efficacy_data[,"cum_cases_placebo"] <- 0
vaccine_efficacy_data[,"cum_cases_vaccine"] <- 0
vaccine_efficacy_data[,"persontime_placebo"] <- 0
vaccine_efficacy_data[,"persontime_vaccine"] <- 0
vaccine_efficacy_data[,"N_placebo_start"] <- 0
vaccine_efficacy_data[,"N_vaccine_start"] <- 0

#calculate follow-up time per-period in each unique study
#loop through studies within each stratum
for(m in unique(vaccine_efficacy_data[, mortality])){
  vaccine_efficacy_stratum <- vaccine_efficacy_data[mortality == m]
  vaccine_efficacy_stratum[,"study_id"] <- cumsum(
    vaccine_efficacy_stratum[,period] == "Period 1"
  )
  for(s in unique(vaccine_efficacy_stratum[, study_id])){
    vaccine_efficacy_study <- vaccine_efficacy_stratum[study_id == s]
    #loop through study-periods
    for(p in 1:nrow(vaccine_efficacy_study)){
      if(p == 1){
        vaccine_efficacy_study[p,"follow_up_p"] <- vaccine_efficacy_study[p,follow_up_end]
        vaccine_efficacy_study[p,"cum_cases_placebo"] <- vaccine_efficacy_study[p,cases_placebo]
        vaccine_efficacy_study[p,"cum_cases_vaccine"] <- vaccine_efficacy_study[p,cases_vaccine]
      } else {
        vaccine_efficacy_study[p,"follow_up_p"] <- vaccine_efficacy_study[p,follow_up_end]
        vaccine_efficacy_study[p,"cum_cases_placebo"] <- vaccine_efficacy_study[p-1,cum_cases_placebo] +
        vaccine_efficacy_study[p,"cum_cases_vaccine"] <- vaccine_efficacy_study[p-1,cum_cases_vaccine] +
      }
    }
    vaccine_efficacy_study[,"N_placebo_start"] <- vaccine_efficacy_study[1,N_placebo]
    vaccine_efficacy_study[,"N_vaccine_start"] <- vaccine_efficacy_study[1,N_vaccine]
```

```
    #calculate follow-up time, use N at start in denominator, as studies do not consistently censor at
    vaccine_efficacy_study[,"persontime_placebo"] <- vaccine_efficacy_study[,N_placebo_start] * vaccine
    vaccine_efficacy_study[,"persontime_vaccine"] <- vaccine_efficacy_study[,N_vaccine_start] * vaccine

    vaccine_efficacy_stratum <- rbindlist(
      list(
        vaccine_efficacy_stratum[study_id != s],
        vaccine_efficacy_study
      )
    )
  }

  vaccine_efficacy_data <- rbindlist(
    list(
      vaccine_efficacy_data[mortality != m],
      vaccine_efficacy_stratum
    )
  )
}

#add 0.5 if no cases have been observed in vaccinated arm (cannot divide by 0), validated method
vaccine_efficacy_data[cum_cases_vaccine == 0, "N_placebo_start"] <- vaccine_efficacy_data[cum_cases_vac
vaccine_efficacy_data[cum_cases_vaccine == 0, "N_vaccine_start"] <- vaccine_efficacy_data[cum_cases_vac
vaccine_efficacy_data[cum_cases_vaccine == 0, "cases_placebo"] <- vaccine_efficacy_data[cum_cases_vacci
vaccine_efficacy_data[cum_cases_vaccine == 0, "cases_vaccine"] <- vaccine_efficacy_data[cum_cases_vacci
vaccine_efficacy_data[cum_cases_vaccine == 0, "cum_cases_placebo"] <- vaccine_efficacy_data[cum_cases_v
vaccine_efficacy_data[cum_cases_vaccine == 0, "cum_cases_vaccine"] <- vaccine_efficacy_data[cum_cases_v

#calculate the relative risk (relative rate not possible with this data)
vaccine_efficacy_data[,"rr"] <- (
  vaccine_efficacy_data[,cum_cases_vaccine]/vaccine_efficacy_data[,N_vaccine_start]
)/(
  vaccine_efficacy_data[,cum_cases_placebo]/vaccine_efficacy_data[,N_placebo_start]
)
vaccine_efficacy_data[,"rr_high"] <- exp(
  log(vaccine_efficacy_data[,rr])
  +1.96*sqrt(
    (
      (vaccine_efficacy_data[,N_vaccine_start] - vaccine_efficacy_data[,cum_cases_vaccine])/vaccine_effi
    )/vaccine_efficacy_data[,N_vaccine_start] +(
      (vaccine_efficacy_data[,N_placebo_start] - vaccine_efficacy_data[,cum_cases_placebo])/vaccine_effi
    )/vaccine_efficacy_data[,N_placebo_start]
  )
)

vaccine_efficacy_data[,"rr_low"] <- exp(
  log(vaccine_efficacy_data[,rr])
  -1.96*sqrt(
    (
      (vaccine_efficacy_data[,N_vaccine_start] - vaccine_efficacy_data[,cum_cases_vaccine])/vaccine_effi
    )/vaccine_efficacy_data[,N_vaccine_start] +(
      (vaccine_efficacy_data[,N_placebo_start] - vaccine_efficacy_data[,cum_cases_placebo])/vaccine_effi
    )/vaccine_efficacy_data[,N_placebo_start]
```

```
  )
)

vaccine_efficacy_data[, "vaccine_efficacy"] <- "cumulative"
vaccine_efficacy_data[, "mortality_factor"] <- factor(
  vaccine_efficacy_data[, mortality],
  # levels=c("Low", "Medium", "High")
  levels=c("VE_data", "iVE_data")
)
```

The models are then fit using rjags.

The waning functions used in the fittings are below in the rjags languge.

```
library(rjags)
#show extracted data by mortality stratum
#plotEstimates(vaccine_efficacy_data)

#load models for pooled analysis in jcodes object (in jags language)

jcodes <- list(
"waning_none"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
        # cases poisson distributed
        cases_placebo[i] ~ dpois(cases_placebo_dist[i])
        cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

        log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
        log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + study_specific_rr[ID

        #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(u in 1:length(uID)){
        study_specific_rr[u] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)
}",
 "waning_linear"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
        # cases poisson distributed
        cases_placebo[i] ~ dpois(cases_placebo_dist[i])
        cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

        log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
        log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + log( exp(study_speci
```

```
        #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(ID in 1:length(uID)){
        study_specific_rr[ID] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    alpha ~ dnorm(0,0.01)
}",
"waning_power"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
    # cases poisson distributed
    cases_placebo[i] ~ dpois(cases_placebo_dist[i])
    cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

    log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
    log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + study_specific_rr[ID[i]]

    #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(ID in 1:length(uID)){
        study_specific_rr[ID] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    alpha ~ dnorm(0,0.01)
}",
"waning_power_01"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
        # cases poisson distributed
        cases_placebo[i] ~ dpois(cases_placebo_dist[i])
        cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

        log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
        log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + log((exp(alpha)*time
```

```
        #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(ID in 1:length(uID)){
        study_specific_rr[ID] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    alpha ~ dnorm(0,0.01)
    beta ~ dnorm(0,0.01)
}",
"waning_power2"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
        # cases poisson distributed
        cases_placebo[i] ~ dpois(cases_placebo_dist[i])
        cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

        log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
        log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + study_specific_rr[ID

        #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(ID in 1:length(uID)){
        study_specific_rr[ID] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    alpha ~ dnorm(0,0.01)
}",
"waning_power3"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
        # cases poisson distributed
        cases_placebo[i] ~ dpois(cases_placebo_dist[i])
        cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

        log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
```

```
        log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + study_specific_rr[ID

        #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(ID in 1:length(uID)){
        study_specific_rr[ID] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    alpha ~ dnorm(0,0.01)
}",
"waning_sigmoid"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
    # cases poisson distributed
    cases_placebo[i] ~ dpois(cases_placebo_dist[i])
    cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

    log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
    log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + study_specific_rr[ID[i]]

    #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(ID in 1:length(uID)){
        study_specific_rr[ID] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    alpha ~ dnorm(0,0.01)
    beta ~ dnorm(0,0.01)
}",
"waning_sigmoid_01"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
    # cases poisson distributed
    cases_placebo[i] ~ dpois(cases_placebo_dist[i])
    cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])
```

```
    log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
    log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + log( exp(study_specific_

    #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific RR, centered around mean RR
    for(ID in 1:length(uID)){
        study_specific_rr[ID] ~ dnorm(mean_rr,between_study_var)
    }

    #prior for mean RR and between study var
    mean_rr ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    alpha ~ dnorm(0,0.01)
    beta ~ dnorm(0,0.01)
}",
"waning_gamma_01"="model{
    #loop through studyIDs
    for (i in 1:length(ID)){
        # cases poisson distributed
        cases_placebo[i] ~ dpois(cases_placebo_dist[i])
        cases_vaccine[i] ~ dpois(cases_vaccine_dist[i])

        log(cases_placebo_dist[i]) = base_rate[i] + log(person_weeks_placebo[i])
        log(cases_vaccine_dist[i]) = base_rate[i] + log(person_weeks_vaccine[i]) + log( pgamma( timept[

        #estimate rate in each study-period
        base_rate[i] ~ dnorm(0,0.01) #prior for risk in unvaccinated
    }

    #prior study specific mean, centered around mean
    for(ID in 1:length(uID)){
        study_specific_alpha[ID] ~ dnorm(alpha,between_study_var)
    }

    #prior for mean RR and between study var
    alpha ~ dnorm(0,0.01)
    between_study_var <- 1/sd^2
    sd ~ dunif(0,2)

    #priors for reduction in RR by time since vaccination
    beta ~ dnorm(0,0.01)
}"
)
```

The waning functions in r language (same as above but in r) are provided below. Ignore the trace warning messages.

```r
#load models for pooled analaysis in rcodes object (in R language)

rcodes <- list(
  "waning_none" = function(t, mean_rr, alpha, beta){
    return(
      rep(
        exp(mean_rr),
        length(t)
      )
    )
  },
  "waning_linear" = function(t, mean_rr, alpha, beta){
    return(
      exp(mean_rr) + exp(alpha)*t
    )
  },
  "waning_power" = function(t, mean_rr, alpha, beta){
    return(
      exp(mean_rr + exp(alpha)*log(t))
    )
  },
  "waning_power_01" = function(t, mean_rr, alpha, beta){
    return(
      (
        exp(alpha)*t^exp(beta)
      )/(
        exp(mean_rr)+exp(alpha)*t^exp(beta)
      )
    )
  },
  "waning_power2" = function(t, mean_rr, alpha, beta){
    return(
      exp(mean_rr + alpha*log(t))
    )
  },
  "waning_power3" = function(t, mean_rr, alpha, beta){
    return(
      exp(mean_rr + alpha*t)
    )
  },
  "waning_sigmoid" = function(t, mean_rr, alpha, beta){
    return(
      exp( mean_rr ) * (1/( 1 + exp(alpha)*exp( -exp(beta)*t) ) )
    )
  },
  "waning_sigmoid_01" = function(t, mean_rr, alpha, beta){
    return(
      exp(mean_rr)/(
        exp(mean_rr) + exp(alpha)*exp( -exp(beta)*t )
      )
    )
  },
  "waning_gamma_01" = function(t, mean_rr, alpha, beta){
```

```
    return(
      pgamma( t, exp(alpha), exp(beta) )
    )
  }
)
```

The models are then fitted using rjags where the estimated parameters from the model are used to calculate the RR at different time points, in the presence of waning (waning model has to be selected within this code). The cumulative VE is then calculated by 1-RR.

```
#select model (waning_power)
model <- c(
  "waning_none", "waning_linear", "waning_power",
  "waning_power_01", "waning_power2", "waning_power3",
  "waning_sigmoid", "waning_sigmoid_01", "waning_gamma_01"
)[3]

#set mcmc options
mcmc.chains <- 2
mcmc.length <- 10000
thin <- 4 ## change thins to 10 if you have issues getting the P values

#run the models with rjags. This will create 3 data-tables: posterior_low,
# posterior_medium, and posterior_high (one for each mortality stratum)
model_code <- jcodes[[model]]
#for(m in c("Low", "Medium", "High")){
for(m in c("VE_data", "iVE_data")){
  jdat <- list(
    cases_placebo=round(vaccine_efficacy_data[mortality == m, cum_cases_placebo]),
    cases_vaccine=round(vaccine_efficacy_data[mortality == m, cum_cases_vaccine]),
    person_weeks_placebo=vaccine_efficacy_data[mortality == m, persontime_placebo],
    person_weeks_vaccine=vaccine_efficacy_data[mortality == m, persontime_vaccine],
    timept=vaccine_efficacy_data[mortality == m, follow_up_end],
    ID=vaccine_efficacy_data[mortality == m, study_id],
    uID=unique(vaccine_efficacy_data[mortality == m, study_id])
  )

  jmodel <- jags.model(
    textConnection(model_code),
    data=jdat,
    n.chains=mcmc.chains,
    n.adapt=1000
  )
  jposterior <- coda.samples(
    jmodel,
    c("mean_rr", "alpha", "beta"),
    n.iter=mcmc.length,
    thin=thin
  )
  #get DIC value
  jdic <- dic.samples(
    jmodel,
    n.iter=mcmc.length,
    thin=thin,
```

```
    type="pD"
  )

  assign(
    paste0("posterior_", tolower(m)),
    jposterior
  )
  assign(
    paste0("dic_", tolower(m)),
    jdic
  )
}
#months in which VE and iVE should be estimated
times <- seq(0.5,36,0.5)

#process modelled results
# generate VE and iVE for each saved draw from the joint posterior distribution)
# see Appendix A in the main paper for more information about the conversion process
# this will create 7 data-tables:
# 3 tables with cumulative VE for each stratum, i.e.: ve_cumulative_low
# 3 tables with instantaneous VE for each stratum, i.e.: ve_instantaneous_low
# 1 table with all VE (iVE and VE) combined: vaccine_efficacy_central


for(m in c("VE_data", "iVE_data")){
  #combine chains
  jposterior <- do.call(
    rbind,
    get(
      paste0("posterior_", tolower(m))
    )
  )
  veCumulative <- rcodes[[model]]

  ve_cumulative <- matrix(
    0,
    ncol = length(times),
    nrow = nrow(jposterior)
  )
  ve_instantaneous <- ve_cumulative
  for(r in 1:nrow(jposterior)){
    ve_cumulative[r, ] <- 1 - veCumulative(
      t = times,
      mean_rr = jposterior[r, "mean_rr"],
      alpha = if("alpha" %in% colnames(jposterior)){
        jposterior[r, "alpha"]
      } else {
        0
      },
      beta = if("beta" %in% colnames(jposterior)){
        jposterior[r, "beta"]
      } else {
        0
```

```
    }
  )
  #convert to instantaneous measure, assume no seasonality in average baseline rate
  ve_instantaneous[r, ] <- cumVEtoInsFast(
    ve_cumulative[r, ],
    0
  )
}

vaccine_efficacy_central <- rbindlist(
  list(
    data.table(
      time = times,
      vaccine_efficacy = rep(
        "cumulative",
        length(times)
      ),
      median = sapply(
        times+1,
        function(x){
          median(
            ve_cumulative[, x]
          )
        }
      ),
      low = sapply(
        times+1,
        function(x){
          quantile(
            ve_cumulative[, x],
            0.025
          )
        }
      ),
      high = sapply(
        times+1,
        function(x){
          quantile(
            ve_cumulative[, x],
            0.975
          )
        }
      )
    ),
    data.table(
      time = times,
      vaccine_efficacy = rep(
        "instantaneous",
        length(times)
      ),
      median = sapply(
        times+1,
        function(x){
```

```
            median(
              ve_instantaneous[, x]
            )
          }
        ),
        low = sapply(
          times+1,
          function(x){
            quantile(
              ve_instantaneous[, x],
              0.025
            )
          }
        ),
        high = sapply(
          times+1,
          function(x){
            quantile(
              ve_instantaneous[, x],
              0.975
            )
          }
        )
      )
    )
  )
  vaccine_efficacy_central[, "mortality"] <- m
  assign(
    paste0(
      "vaccine_efficacy_central_",
      tolower(m)
    ),
    vaccine_efficacy_central
  )
  assign(
    paste0(
      "ve_cumulative_",
      tolower(m)
    ),
    ve_cumulative
  )
  assign(
    paste0(
      "ve_instantaneous_",
      tolower(m)
    ),
    ve_instantaneous
  )
}
vaccine_efficacy_central <- rbindlist(
  list(
    vaccine_efficacy_central_ive_data,
    # vaccine_efficacy_central_medium,
```

```
    vaccine_efficacy_central_ve_data
  )
)

vaccine_efficacy_central[, "mortality_factor"] <- factor(
  vaccine_efficacy_central[, mortality],
  levels=c("VE_data", "iVE_data")
)
```

Plots

Plots for each waning function and for different methods of calculating the vaccine efficacy. Below is for one waning function only - these need to be changed within the code above where it states:

select model (waning_power) model <- c("waning_none", "waning_linear", "waning_power", "waning_power_01", "waning_power2", "waning_power3", "waning_sigmoid", "waning_sigmoid_01", "waning_gamma_01")[3]

Waning (Power) - "waning_power" has been selected in the above example.

```
vaccine_efficacy_data_VE <- vaccine_efficacy_data[ which(vaccine_efficacy_data$mortality=='VE_data'), ]
vaccine_efficacy_data_iVE <- vaccine_efficacy_data[ which(vaccine_efficacy_data$mortality=='iVE_data'),
vaccine_efficacy_central_VE <- vaccine_efficacy_central[ which(vaccine_efficacy_central$mortality=='VE_
vaccine_efficacy_central_iVE <- vaccine_efficacy_central[ which(vaccine_efficacy_central$mortality=='iV
vaccine_efficacy_central_iVE_calc <- vaccine_efficacy_central[ which(vaccine_efficacy_central$mortality

VEdat <- ggplot()+
  geom_errorbar(data=vaccine_efficacy_data_VE, aes(x=follow_up_end, ymin=1-rr_high, ymax=1-rr_low))+
  geom_point(data=vaccine_efficacy_data_VE, aes(x=follow_up_end, y=1-rr, size=(cases_vaccine + cases_pla
  coord_cartesian(ylim=c(0, 1), xlim=c(0.2, 36))+
  ggtitle("Cumulative Vaccine Efficacy") +
  labs(x="", y="Vaccine efficacy (prop)\n")+
  scale_size_continuous(range = c(0.5,2.5))+
  guides(size="none")+
  geom_hline(yintercept=0)


if(!is.null(vaccine_efficacy_central_VE)){
  VEdat <- VEdat +geom_ribbon(data=vaccine_efficacy_central_VE,aes(x=time, ymin=low, ymax=high), alpha=
    geom_line(data=vaccine_efficacy_central_VE,aes(x=time,y=median))+
    theme_bw()+
    theme(axis.text=element_text(size=10),
          axis.title=element_text(size=14,face="bold"),
          legend.text = element_text(size=10),
          legend.title = element_text(size=12),
          plot.title = element_text(colour = "grey39"))
}


##### Instantaneous Vaccine Efficacy #####

iVEdat <- ggplot()+
  geom_errorbar(data=vaccine_efficacy_data_iVE, aes(x=follow_up_end, ymin=1-rr_high, ymax=1-rr_low))+
  geom_point(data=vaccine_efficacy_data_iVE, aes(x=follow_up_end, y=1-rr, size=(cases_vaccine + cases_pl
```

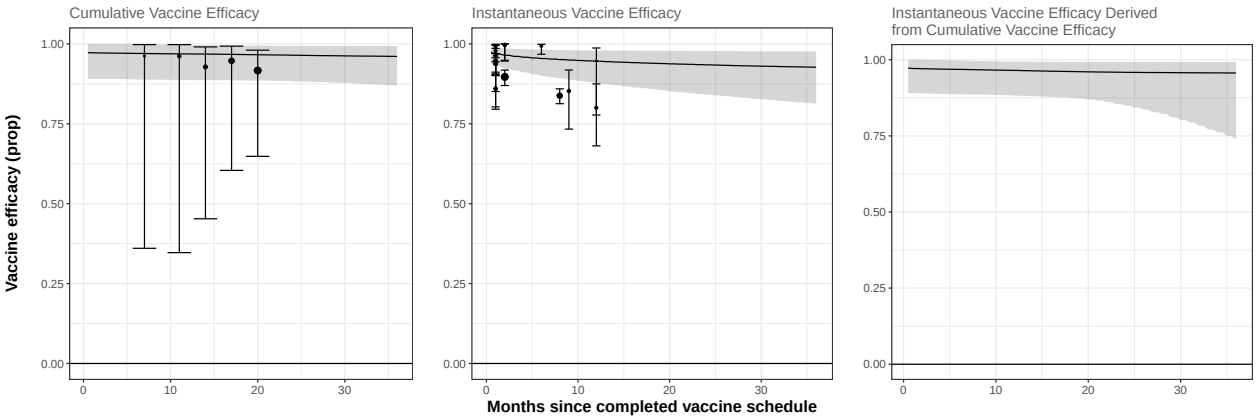

Figure 1: Figure: Comparisons of vaccine efficacy by month since vaccine schedule was completed

```
  coord_cartesian(ylim=c(0, 1), xlim=c(0.2, 36))+
  ggtitle("Instantaneous Vaccine Efficacy") +
  #facet_grid(vaccine_efficacy~mortality_factor)+
  labs(x="Months since completed vaccine schedule", y="")+
  scale_size_continuous(range = c(0.5,2.5))+
  guides(size="none")+
  geom_hline(yintercept=0)


if(!is.null(vaccine_efficacy_central_iVE)){
  iVEdat <- iVEdat +geom_ribbon(data=vaccine_efficacy_central_iVE,aes(x=time, ymin=low, ymax=high), alp
    geom_line(data=vaccine_efficacy_central_iVE,aes(x=time,y=median))+
    theme_bw()+
    theme(axis.text=element_text(size=10),
          axis.title=element_text(size=14,face="bold"),
          legend.text = element_text(size=10),
          legend.title = element_text(size=12),
          plot.title = element_text(colour = "grey39"))
}

##### Instantaneous Vaccine Efficacy Calculated from cumulative VE #####

iVEdat_cal <- ggplot()+
geom_ribbon(data=vaccine_efficacy_central_iVE_calc,aes(x=time, ymin=low, ymax=high), alpha=0.2)+
  geom_line(data=vaccine_efficacy_central_iVE_calc,aes(x=time,y=median))+
  ggtitle("Instantaneous Vaccine Efficacy Derived\nfrom Cumulative Vaccine Efficacy") +
  theme_bw()+
  labs(x="", y="")+
  geom_hline(yintercept=0)+
  theme(axis.text=element_text(size=10),
        axis.title=element_text(size=14,face="bold"),
        legend.text = element_text(size=10),
        legend.title = element_text(size=12),
        plot.title = element_text(colour = "grey39"))

grid.arrange(VEdat,iVEdat,iVEdat_cal, ncol=3)
```