Group Randomized Trial Design For Targeted Agent – A Simulation Study

Zijiang Yang

March 30, 2017

```
library(powerSurvEpi)
library(stats)
library(base)
library(survival)
library(PwrGSD)
library(ggplot2)
```

The simulation

```
simulation <- function(sims,countries,positive_only,assumption,test,T0,T1){
  # sims: How many simulations
  # countries: How many countries
  # positive_only: Include only positive patients or not
  # T0: End of accrural (in year)
  # T1: End of study (in year)
  #
  # test: Type of hypothesis testing
  # #"log rank": Perform a log rank test
  # #"t": Perform a one-tailed t test on the five-year survival rate
  # assumption: ssumption of efficacy of the drug
  # #"best": 45% for +, 40% for -
  # #"unif": unif(0.35-0.45) for +, unif(0.3-0.4) for -
  T0 <- 365*T0
  T1 <- 365*T1
  Total_N <- 0
  f <- c()
  significance <- c()
  p_list <- c()
  N <- 100

  for (round in 1:sims){
    if (round%%50 == 0){
      cat("Round ", round, "\n")
    }
    big_dat <- data.frame()
    cur_id <- 1
    cur_time <- 0
    accrural_end <- 0
    i <- 1
    while(accrural_end==0){
      set.seed(round*1001+i)
      x <- rbinom(N,1,0.25) #Biomarker status
      g <- rbinom(N,1,0.5) #Group assignment
```

```
sp_list <- c()
s <- c()
c <- c()
delta <- c() #Censoring status
follow_up <- c() #Actual follow-up days since intake (considering censoring)
five_yr_live <- c() #Alive at five year of follow-up or not
accrued <- N
for (j in 1:N){
  #Simulate entry time in days
  gap <- rexp(1,(72*countries)/365) #S_i-S_{i-1}, assuming 3 countries: 72*3 = 216
  cur_time <- cur_time+gap
  if (cur_time > T0){ #Stop Accrural
    accrued <- j-1
    accrural_end <- 1
    break
  }
  s <- c(s,cur_time)
  #Simulate survival probability of 5 years
  sp <- 0
  if (g[j] == 1){
    if (assumption=="unif"){
      #Uniform Scenario
      if (x[j] == 1) sp <- runif(1,0.35,0.45)
      else sp <- runif(1,0.3,0.4)
    }
    else if (assumption=="best"){
      #Best case scenario
      if (x[j] == 1) sp <- 0.45
      else sp <- 0.4
    }
  }else{
    sp <- 0.3
  }
  sp_list <- c(sp_list,sp)
  #Simulate censoring time since intake (in days)
  c_day <- rexp(1,0.05/365)
  c <- c(c,c_day)
}
if (accrued == 0){
  break
}
lambda <- -log(sp_list)/5
t <- c() #Survival time since intake (in days)
for (j in 1:accrued){
  st <- rexp(1,lambda[j])*365
  t <- c(t,st)
  delta <- c(delta,ifelse(c[j]<=t[j],1,0)) #right censored
  follow_up <- c(follow_up,min(t[j],c[j]))
  if (s[j]+follow_up[j]>T1){
    follow_up[j] <- T1-s[j]
    delta[j] <- 1 #right censored
  }
  five_yr_live <- c(five_yr_live,ifelse(t[j]>=1825,1,0))
```

```
      }
      id <- seq(cur_id,cur_id+accrued-1)
      cur_id <- cur_id+accrued
      if (accrural_end){
        x <- x[1:accrued]
        g <- g[1:accrued]
      }
      dat <- data.frame(id=id,x=x,g=g,s=s,t=t,c=c,delta=delta,follow_up=follow_up,five_yr_live=five_yr_l
      big_dat <- rbind(big_dat,dat)
      i <- i + 1
    }
    if(positive_only){
      sub <- subset(big_dat,x==1)
    }else{
      sub <- big_dat
    }
    Total_N <- Total_N + length(sub$id)
    f <- c(f,sum(sub$follow_up)/length(sub$follow_up))
    if (test=="log rank"){
      s <- survdiff(Surv(follow_up,delta==0) ~ g,data=sub)
      p <- 1 - pchisq(s$chisq, 1)
      p_list <- c(p_list,p)
      #print(p)
      significance <- c(significance,ifelse(p<=0.05,1,0))
    }else if (test == "t"){
      sub <- subset(sub,c>t) #delete dropouts
      trt <- subset(sub,g==1)
      ctr <- subset(sub,g==0)
      #print(length(ctr$five_yr_live))
      s <- t.test(ctr$five_yr_live,trt$five_yr_live,alternative = "less", var.equal = TRUE)
      p_list <- c(p_list,s$p.value)
      significance <- c(significance,ifelse(s$p.value<=0.05,1,0))
    }else{
      print("Please enter correct test type!")
      quit()
    }
  }

  cat("T0 = ", T0, "\n")
  cat("T1 = ", T1, "\n")
  cat("test = ", test, "\n")
  cat("assumption = ", assumption, "\n")
  cat("positive_only = ", positive_only, "\n")
  cat("countries = ", countries, "\n")
  cat("power = ",mean(significance), "\n")
  cat("Mean total sample size = ",Total_N/sims, "\n")
  cat("Mean follow_up time per subject= ",mean(f), "\n\n\n")

  sim <- data.frame(T0=T0,T1=T1,assumption=assumption,positive_only=positive_only,countries=countries,sa
  return (sim)
}

# direct output to a file
```

```r
#sink("/Users/river/Documents/699/Project_4/output.txt", append=TRUE, split=FALSE)
```

```r
sim_results <- data.frame()

for (countries in 3:4){
  for (positive_only in c(TRUE,FALSE)){
    for (assumption in c("best","unif")){
      for (T0 in 3:5){
        for (T1 in 5:10){
          test <- "log rank"
          sims <- 0 #Set to 0 for output purpose only; should be 1000
          sim <- simulation(sims,countries,positive_only,assumption,test,T0,T1)
          sim_results<-rbind(sim_results,sim)
        }
      }
    }
  }
}

#sink()
sim_results$T0 <- sim_results$T0/365
sim_results$T1 <- sim_results$T1/365
```

Simulation Results

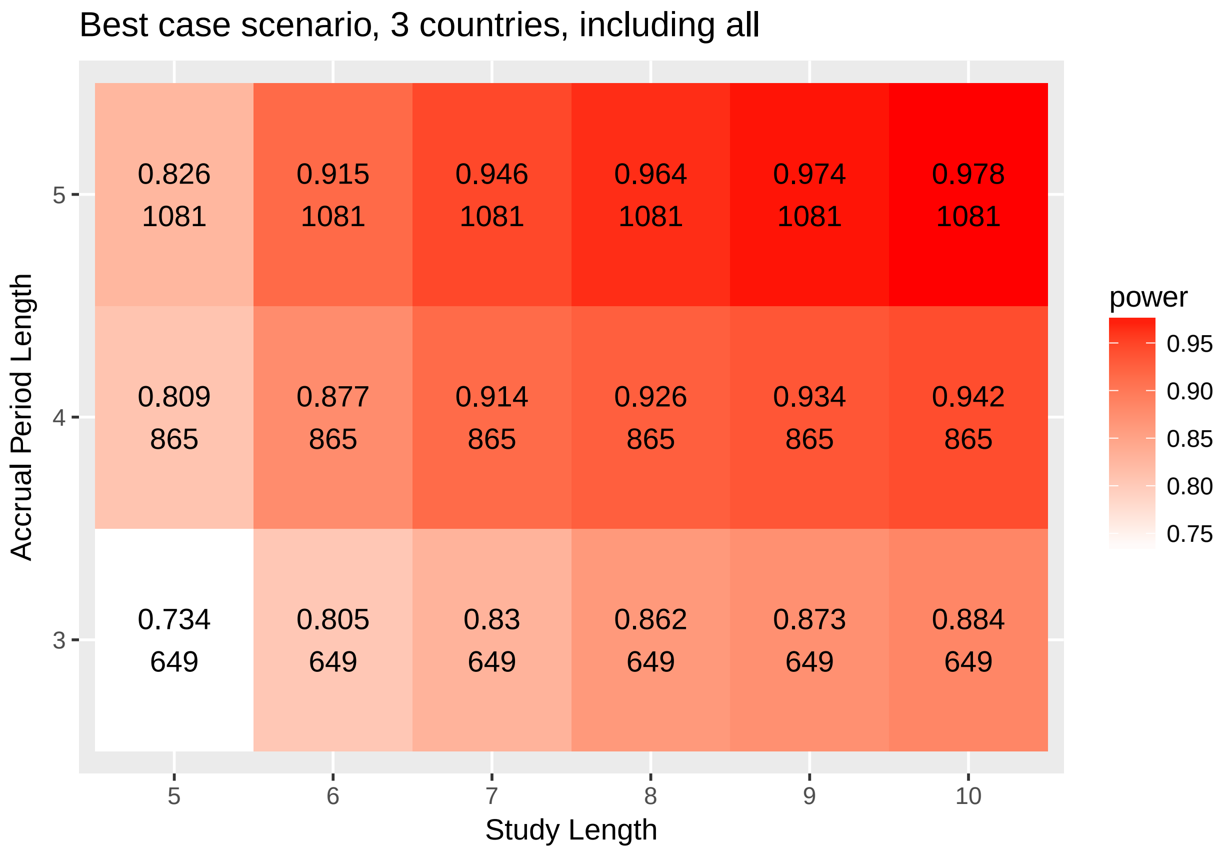
```r
#write.table(sim_results, file = "sim_results.csv", sep = ",", col.names = NA, qmethod = "double")
#For output purpose only, loading saved results from simulation
sim_results <- read.table("/Users/river/Documents/699/Project_4/sim_results.csv",sep=",",header = TRUE)
#Listing results that have power >= 0.7
subset(sim_results,power>=0.7)[,2:9]
```

```
##     T0 T1 assumption positive_only countries sample_size power follow_up
## 12   4 10       best          TRUE         3     216.560 0.713 1262.2956
## 16   5  8       best          TRUE         3     270.299 0.720 1070.3796
## 17   5  9       best          TRUE         3     270.299 0.753 1159.2970
## 18   5 10       best          TRUE         3     270.299 0.787 1229.1874
## 37   3  5       best         FALSE         3     648.561 0.734  826.5350
## 38   3  6       best         FALSE         3     648.561 0.805  960.5547
## 39   3  7       best         FALSE         3     648.561 0.830 1064.6032
## 40   3  8       best         FALSE         3     648.561 0.862 1145.2311
## 41   3  9       best         FALSE         3     648.561 0.873 1207.7733
## 42   3 10       best         FALSE         3     648.561 0.884 1256.2934
## 43   4  5       best         FALSE         3     865.106 0.809  732.7468
## 44   4  6       best         FALSE         3     865.106 0.877  888.0599
## 45   4  7       best         FALSE         3     865.106 0.914 1008.5421
## 46   4  8       best         FALSE         3     865.106 0.926 1102.0410
## 47   4  9       best         FALSE         3     865.106 0.934 1174.6468
## 48   4 10       best         FALSE         3     865.106 0.942 1230.9584
## 49   5  5       best         FALSE         3    1080.642 0.826  620.0889
## 50   5  6       best         FALSE         3    1080.642 0.915  800.9620
## 51   5  7       best         FALSE         3    1080.642 0.946  941.0860
```

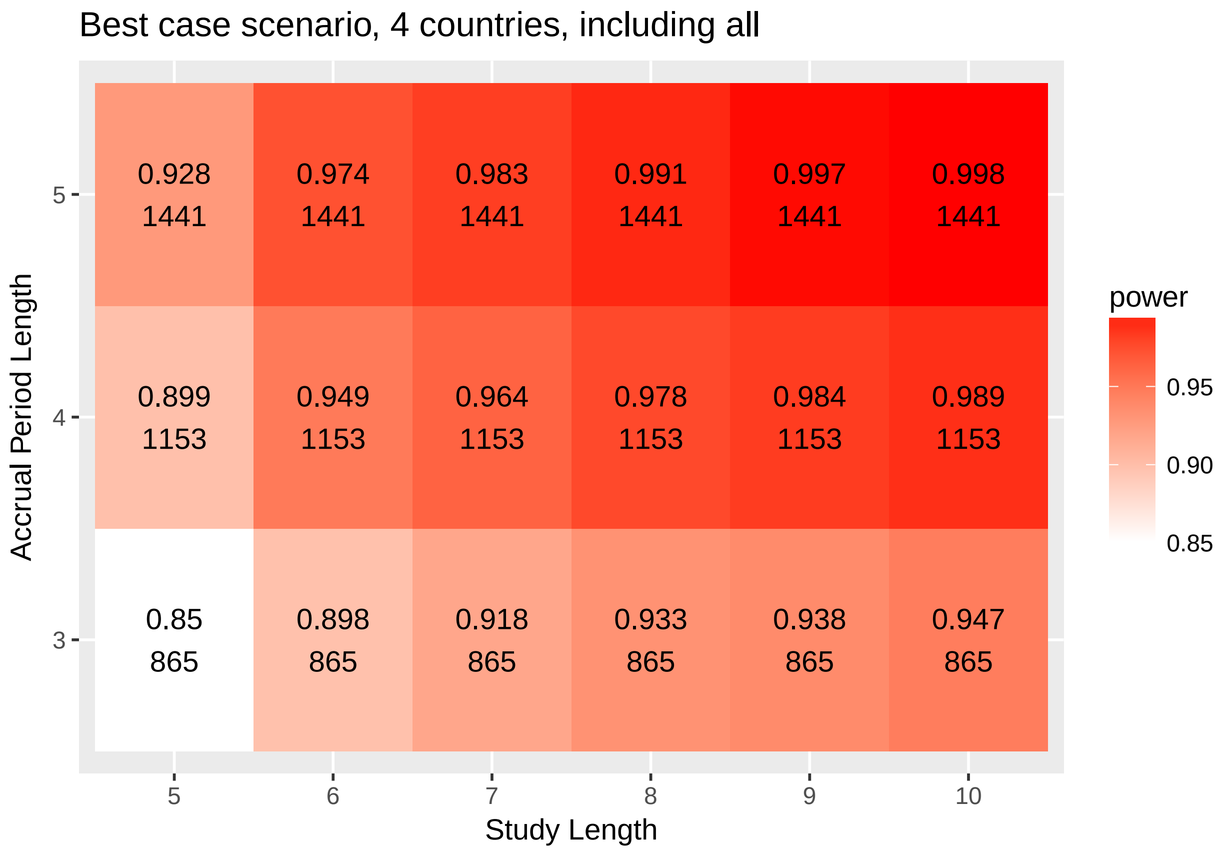
```
## 52    5   8        best          FALSE          3     1080.642 0.964 1049.7834
## 53    5   9        best          FALSE          3     1080.642 0.974 1134.1100
## 54    5  10        best          FALSE          3     1080.642 0.978 1199.5141
## 77    3   9        best           TRUE          4      216.560 0.700 1238.0129
## 78    3  10        best           TRUE          4      216.560 0.713 1290.9061
## 81    4   7        best           TRUE          4      288.530 0.745 1026.9596
## 82    4   8        best           TRUE          4      288.530 0.772 1125.3156
## 83    4   9        best           TRUE          4      288.530 0.814 1202.5775
## 84    4  10        best           TRUE          4      288.530 0.822 1263.2088
## 86    5   6        best           TRUE          4      360.406 0.729  812.2796
## 87    5   7        best           TRUE          4      360.406 0.796  957.3533
## 88    5   8        best           TRUE          4      360.406 0.842 1070.9712
## 89    5   9        best           TRUE          4      360.406 0.866 1160.1464
## 90    5  10        best           TRUE          4      360.406 0.886 1230.1165
## 109   3   5        best          FALSE          4      865.106 0.850  826.7241
## 110   3   6        best          FALSE          4      865.106 0.898  960.9169
## 111   3   7        best          FALSE          4      865.106 0.918 1065.0849
## 112   3   8        best          FALSE          4      865.106 0.933 1145.9542
## 113   3   9        best          FALSE          4      865.106 0.938 1208.7147
## 114   3  10        best          FALSE          4      865.106 0.947 1257.4264
## 115   4   5        best          FALSE          4     1152.824 0.899  732.9600
## 116   4   6        best          FALSE          4     1152.824 0.949  888.3083
## 117   4   7        best          FALSE          4     1152.824 0.964 1008.8364
## 118   4   8        best          FALSE          4     1152.824 0.978 1102.3353
## 119   4   9        best          FALSE          4     1152.824 0.984 1174.8590
## 120   4  10        best          FALSE          4     1152.824 0.989 1231.0889
## 121   5   5        best          FALSE          4     1441.158 0.928  619.9900
## 122   5   6        best          FALSE          4     1441.158 0.974  800.9211
## 123   5   7        best          FALSE          4     1441.158 0.983  941.1725
## 124   5   8        best          FALSE          4     1441.158 0.991 1049.8935
## 125   5   9        best          FALSE          4     1441.158 0.997 1134.2076
## 126   5  10        best          FALSE          4     1441.158 0.998 1199.5456
## 138   4  10        unif          FALSE          4     1150.819 0.706 1183.4834
## 142   5   8        unif          FALSE          4     1438.970 0.718 1017.7893
## 143   5   9        unif          FALSE          4     1438.970 0.764 1095.4418
## 144   5  10        unif          FALSE          4     1438.970 0.794 1154.7509
```

```
best_3_f <- subset(sim_results,assumption=="best" & countries==3 & positive_only==FALSE)
unif_4_f <- subset(sim_results,assumption=="unif" & countries==4 & positive_only==FALSE)
best_4_f <- subset(sim_results,assumption=="best" & countries==4 & positive_only==FALSE)
best_4_t <- subset(sim_results,assumption=="best" & countries==4 & positive_only==TRUE)
```

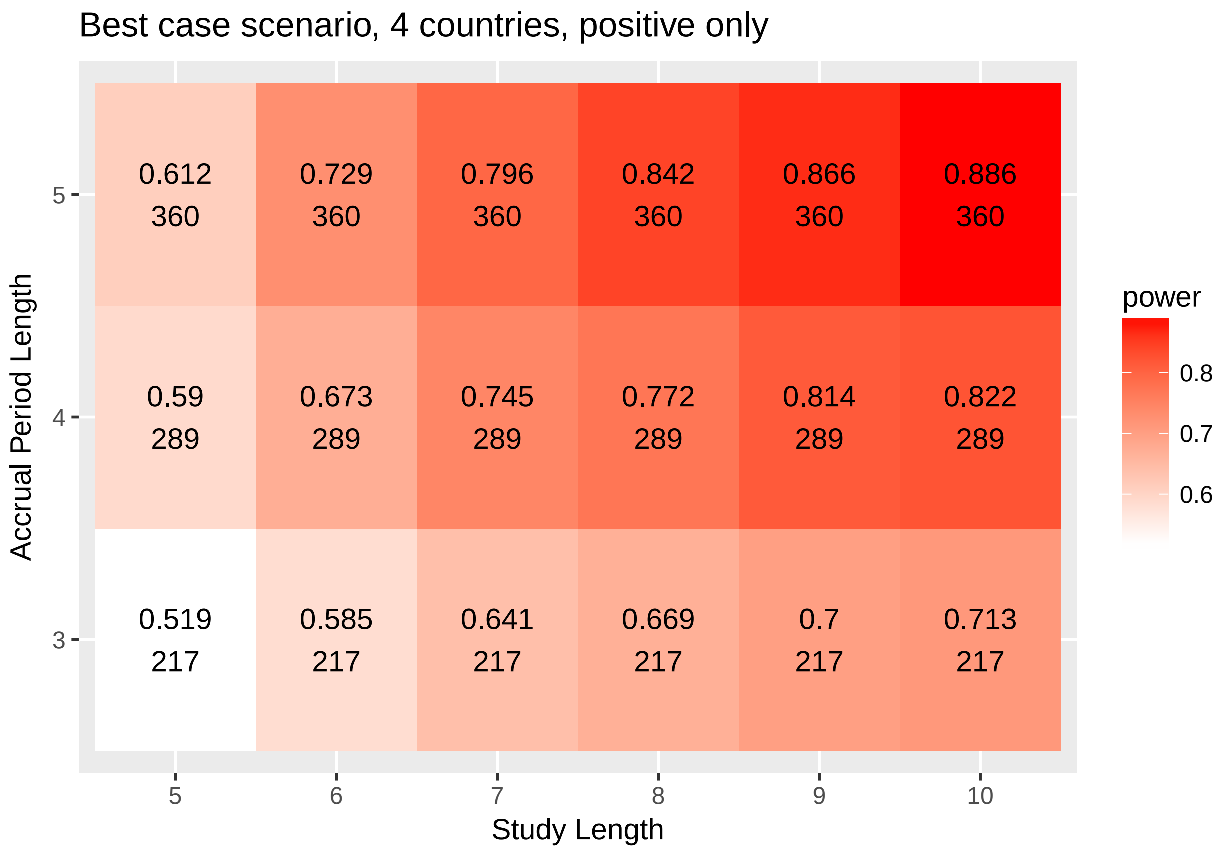
```
ggplot(best_3_f, aes(as.factor(T1), as.factor(T0))) +
    geom_tile(aes(fill = power)) +
    geom_text(aes(label = paste(power,round(sample_size), sep="\n"))) +
    scale_fill_gradient(low = "white", high = "red") +
    ggtitle("Best case scenario, 3 countries, including all") +
    labs(x="Study Length",y="Accrual Period Length")
```



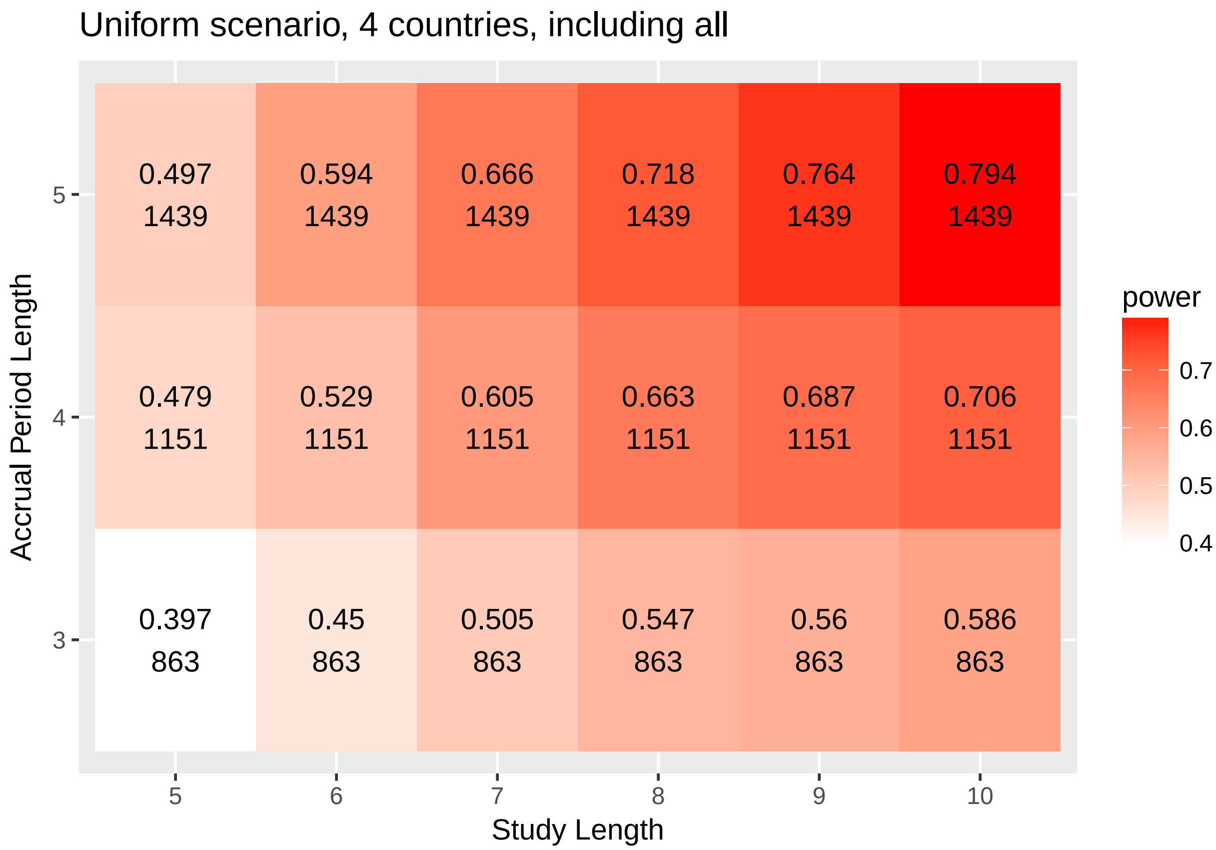

```r
ggplot(best_4_f, aes(as.factor(T1), as.factor(T0))) +
    geom_tile(aes(fill = power)) +
    geom_text(aes(label = paste(power,round(sample_size), sep="\n"))) +
    scale_fill_gradient(low = "white", high = "red") +
    ggtitle("Best case scenario, 4 countries, including all") +
    labs(x="Study Length",y="Accrual Period Length")
```




```r
ggplot(best_4_t, aes(as.factor(T1), as.factor(T0))) +
    geom_tile(aes(fill = power)) +
    geom_text(aes(label = paste(power,round(sample_size), sep="\n"))) +
    scale_fill_gradient(low = "white", high = "red") +
    ggtitle("Best case scenario, 4 countries, positive only") +
    labs(x="Study Length",y="Accrual Period Length")
```




```r
ggplot(unif_4_f, aes(as.factor(T1), as.factor(T0))) +
    geom_tile(aes(fill = power)) +
    geom_text(aes(label = paste(power,round(sample_size), sep="\n"))) +
    scale_fill_gradient(low = "white", high = "red") +
    ggtitle("Uniform scenario, 4 countries, including all") +
    labs(x="Study Length",y="Accrual Period Length")
```


Uniform scenario, 4 countries, including all
0.497 1439
0.594 1439
0.666 1439
0.718 1439
0.764 1439
0.794 1439
0.479 1151
0.529 1151
0.605 1151
0.663 1151
0.687 1151
0.706 1151
0.397 863
0.45 863
0.505 863
0.547 863
0.56 863
0.586 863
Accrual Period Length
5
4
3
Study Length
5
6
7
8
9
10
power
0.7
0.6
0.5
0.4