HW5: Power analsys and Permutation P-values

STAT-COMP

Due Dec. 2 in D2L

Question 1: Power analysis in the Logistic Regression model

Consider a simple logistic regression model of the form

$log(\theta_i/(1-\theta_i)) = \mu + X_i\beta$

where θ_i is the probability of disease of an individual and X_i is a dummy variable for a treatment (e.g., Covid-19 vaccination).

In the above model, the log of the odds of disease is equal to μ in the control group ($X_i = 0$) and to $\mu + \beta$ in the vaccinated group. Here, $\beta < 0$ indicates that vaccination is effective at reducing the risk of developing disease.

Power in this model depends on:

- the effect size (i.e., the OR),
- sample size,
- the prevalence in the control group, and
- the propotion of individuals in the trial that are vaccinated.

Our goal is to estimate power for possible values of the first three factors.

The following code can be used to simulate binary data for the above model using a sample size of 1,000 for a case-control trial with equal number of vaccinated and unvaccinated people, a prevalence of disease in the control group of 0.1, and an odds-ratio of 0.85 (i.e., vaccination is protective).

```
set.seed(1950)
n=1000
prev=0.1
OR=0.85

X=(runif(n)<0.5)*1.0 # dummy variable for vaccination
b=log(OR) #effect size as a function of the odds-ratio
mu=log(prev/(1-prev))
ETA=mu+X*b

# probability of disease
predProb=exp(ETA)/(1+exp(ETA))

# sampling the outcome
y=1.0*(runif(n)<predProb)
table(X,y)
```

```
##    y
## X     0   1
##   0 445  54
##   1 452  49
```

For this simualted data set a logistic regression model yields the following results

```
fm0=glm(y~X,family='binomial')
summary(fm0)
```

```
##
## Call:
## glm(formula = y ~ X, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4786  -0.4786  -0.4537  -0.4537   2.1563
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.1091     0.1441 -14.636   <2e-16 ***
## X            -0.1128     0.2083  -0.541    0.588
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 663.25  on 999  degrees of freedom
## Residual deviance: 662.96  on 998  degrees of freedom
## AIC: 666.96
##
## Number of Fisher Scoring iterations: 4
```

Note that the pvalue for the treatment effect in the previous analysis is for a two-sided test. In this case, we want to test weather vacciation is protective, to make it simple, consider H_0: $\beta = 0$ Vs H_a: $\beta < 0$. A pvalue for that one-sided test can be obatined using

```
tStat=summary(fm0)$coef[2,3]
pVal=pnorm(tStat,lower.tail=TRUE)
pVal
```

```
## [1] 0.2941005
```

Task: Modify the code presented above to estimate power for each of the following scenarios.

```
SCEN=expand.grid(n=c(300,1000,3000,10000),OR=c(0.9,0.8,0.7,0.6),prevalence=c(.1,.2),power=NA)
```

To determine power:

```
- Use a one-sided test pvalue, rejecting if that pvalue is <0.05
- Use at least 1,000 MC replicates per scenario.
```

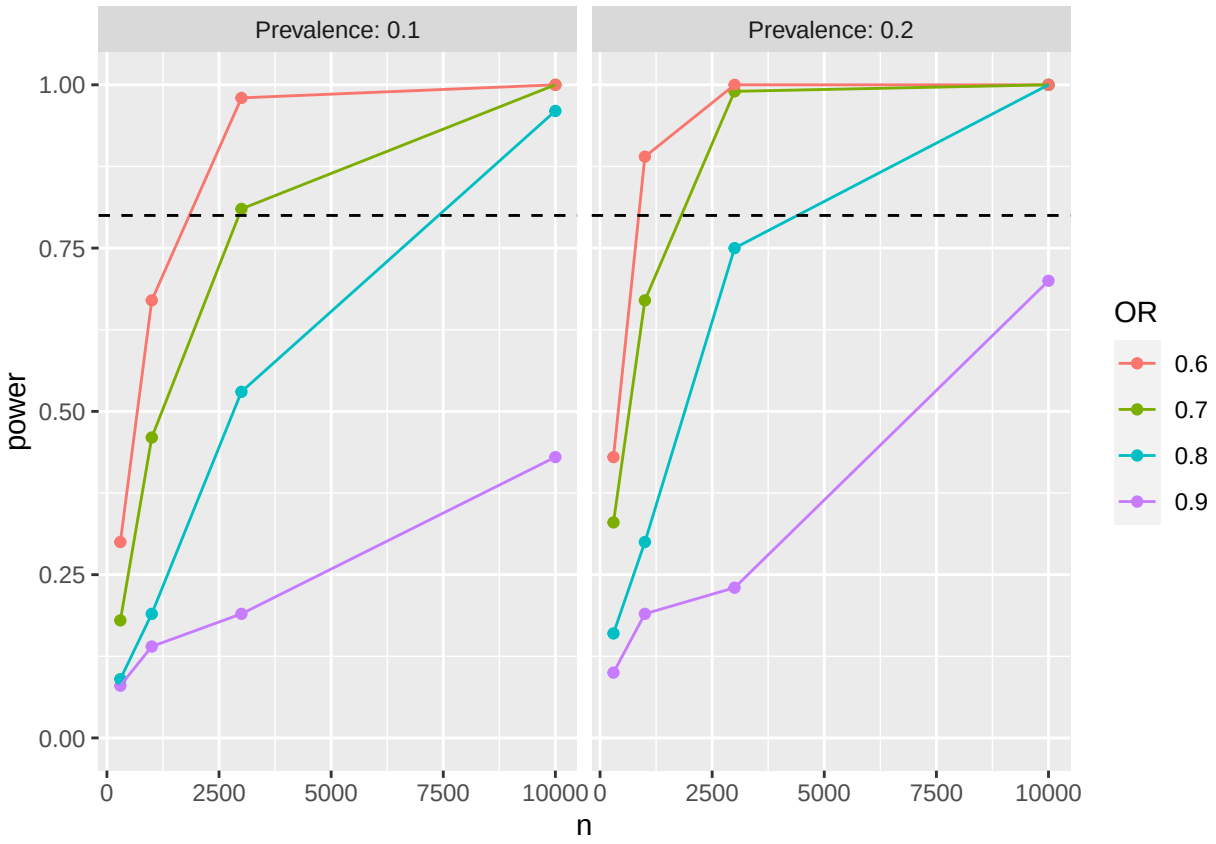
Present a power plot (see template code below) and report the sample size you would recommend for each of these scenarios for a power of at least 0.8.

OR	Prevalence in the control group	Recommended Sample Size
0.8	0.1	
0.7	0.1	
0.8	0.2	
0.7	0.2	

```
nReps=100
for(scen in 1:nrow(SCEN)){

 n=SCEN$n[scen]
 OR=SCEN$OR[scen]
 prev=SCEN$prevalence[scen]
 prop_exposed=SCEN$prop_exposed[scen]

 pValues=rep(NA,nReps)
 n0=round(n/2)


 for(i in 1:nReps){
  # A dummy variable for exposure

   X=c(rep(0,n0),rep(1,n-n0))

   mu=log(prev/(1-prev))
   b=log(OR)

   ETA=mu+X*b

   predProb=exp(ETA)/(1+exp(ETA))

   y=(runif(n)<predProb)*1.0 # same as rbinom(size=1,n=n,...)

   fm=glm(y~X,family='binomial')

   pValues[i]=pnorm(summary(fm)$coef[2,3],lower.tail=TRUE)
  }

  SCEN$power[scen]=mean(pValues<0.05)

 }
```

Template code to generate the power-plot

```
library(ggplot2)
SCEN$OR=factor(SCEN$OR)
SCEN$prevalence=factor(paste0('Prevalence: ',SCEN$prevalence))
p=ggplot(SCEN,aes(x=n,y=power))+
    geom_point(aes(color=OR))+
    geom_line(aes(color=OR))+
    ylim(c(0,1))+
    facet_grid(~prevalence)+
    geom_hline(yintercept = .8,linetype='dashed')#+
    #geom_vline(xintercept = c(2000,3000,4000,7500),linetype='dashed')
plot(p)
```



2 Permuation pvalues

We can use permutations to estimate pvalues. This can be done using the following algorithm:

(i) Permute the data (either the covariate or the response)
(ii) Fit the model to the permuted data and extract the parameter estimate (e.g., the regression coefficient), store it in a vector.
(iii) Repeat (i) and (ii) a very large number of times.
(iv) Find the proportion of times the permutation estimates are more extreme than the estimate you obtained in the un-permuted data.

Task: Use 10,000 permutations to estimate the p-value for a one-sided test (H_0: $\beta = 0$ Vs H_a: $\beta < 0$) on the effect of vaccination using this data:

```
set.seed(1950)
n=1000
prev=0.1
OR=0.9

X=(runif(n)<0.5)*1.0 # dummy variable for vaccination
b=log(OR) #effect size as a function of the odds-ratio
mu=log(prev/(1-prev))
ETA=mu+X*b

# probability of disease
predProb=exp(ETA)/(1+exp(ETA))

# sampling the outcome
```

```r
y=1.0*(runif(n)<predProb)
fm0=glm(y~X,family='binomial')
```

Compare your permutation pvalue with the conventional pvalue for a one-sided test.

```r
permEstimates=rep(NA,10000)
for(i in 1:10000){
  Z=sample(X,size=length(X),replace=FALSE)
  fm=glm(y~Z,family='binomial')
  permEstimates[i]=coef(fm)[2]
}
permPval=mean(permEstimates<coef(fm0)[2])
conventionalPval=pnorm(summary(fm0)$coef[2,3],lower.tail=TRUE)

c(conventionalPval,permPval)
```

```
## [1] 0.4101245 0.3977000
```