survSNP: Power and Sample Size Calculations for SNP Association Studies with Censored Time to Event Outcomes

Kouros Owzar Zhiguo Li Nancy Cox Sin-Ho Jung
Chanhee Yi

January 18, 2015

1 Introduction

This vignette serves as a tutorial for using the survSNP extension package for conducting power and sample size calculations for SNP association studies with censored time to event outcomes. We begin by loading the package.

```
library(survSNP)
```

```
## Loading required package: survival
## Loading required package: splines
## Loading required package: Rcpp
## Loading required package: lattice
## Loading required package: foreach
## Loading required package: xtable
```

2 Example 1

In this example, we conduct power calculations for a SNP for which the relative allelic frequency for the risk allele B is $q = 0.1$. The validation data set is assumed to consist of $n = 500$ patients. The event rate (e.g., death rate) is assumed to be $\rho = 0.75$ (i.e., 375 events among 500 patients). We hypothesize an effect size (genotype hazard ratio) $\Delta = 1.25$. Finally, we assume that the median in the population is 1 unit of time (i.e., $P(\tilde{T} > t) = 0.5$).

To find the asymptotic power, at the two-sided $\alpha = 0.05$ level, the `sim.snp.expsurv.power` function is used.

```
res1<-sim.snp.expsurv.power(1.25, n=500, raf=0.1, erate=0.75,
                            pilm=0.5, lm=1, B=0,
                            model="additive",test="additive",alpha=0.05)
```

Below, we show some of the relevant columns from the output.

```
res1[,c("n","GHR","erate","raf","B","alpha","pow0","pow","powB")]
```

```
##         n  GHR erate raf B alpha      pow0 pow powB
## power 500 1.25  0.75 0.1 0  0.05 0.4706994  NA   NA
```

The asymptotic power, at the two-sided $\alpha = 0.05$ level, is 0.47. Note that we are assuming that the median for the survival function in the population is 1 unit of time (that is why we have set `pilm=0.5` and `lm=1`). If we had desired to set power the study based on a population whose 0.7 quantile is say 2 units of time, we would have set `pilm=0.7` and `lm=2`. The current version of the package supports power calculations for additive models.

By default, the asymptotic power based on the approximate asymptotic variance formula is computed. The corresponding column name is `pow0` in the output shown above. The asymptotic power based on the exact variance formula (column `pow`) or the empirical power (column `powB`) can be computed by setting the arguments `exactvar=TRUE` or `B=b` where `b` is a positive integer. Be sure to set a random number generation seed when calculating thed empirical or the asymptotic power based on the exact variance formula to get reproducible results. We illustrate these features next. This example is based on $B = 10$ simulation repplicates.

```
set.seed(123)
res1b<-sim.snp.expsurv.power(1.25, n=500, raf=0.1, erate=0.75,
                            pilm=0.5, lm=1,
                            exactvar=TRUE,B=B,
                            model="additive",test="additive",alpha=0.05)
```

```
## Warning in coxph(Surv(otime, event) ~ SNP): X matrix deemed to be singular;
variable 2

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.
```

```
## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
```

```
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.
```

```
## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in coxph(Surv(otime, event) ~ SNP): X matrix deemed to be singular;
variable 2

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
```

```
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.
```

```
## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
```

```
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.

## Warning in fitter(X, Y, strats, offset, init, control, weights = weights, :
Loglik converged before variable  2 ; beta may be infinite.
```

The results are shown below.

```
res1b[,c("n","GHR","erate","raf","B","alpha","pow0","pow","powB")]


##         n  GHR erate raf     B alpha      pow0       pow   powB
## power 500 1.25  0.75 0.1 10000  0.05 0.4706994 0.4667144 0.4315
```

3 Example 2

For power calculations for SNP association studies with censored outcomes, it is generally desired to vary the effect size, sample size, event rate and the relative minor allele frequency. These are denoted by the variable names `GHRs`, `ns`, `rafs` and `erates` in this example.

```
GHRs<-seq(1.05,1.5,by=0.05)
ns<-c(100,500,700)
rafs<-c(0.1,0.3,0.5)
erates<-c(0.5,0.7,0.9)
```

The function `survSNP.power.table` can be used to generate power calculations for this combination of paramaters. This is a wrapper function for `sim.snp.expsurv.power`.

```
res2<-survSNP.power.table(GHRs,ns,rafs,erates,
                          pilm=0.5,lm=1,
                          model="additive",test="additive",
                          alpha=0.05)
```

We print selected columns from the first three rows of the previous object next.

```
res2[1:3,c("n","GHR","erate","raf","pow0","pow","powB")]
```

```
##          n  GHR erate raf       pow0 pow powB
## power  100 1.05   0.5 0.1 0.05719290  NA   NA
## power1 100 1.05   0.7 0.1 0.05819468  NA   NA
## power2 100 1.05   0.9 0.1 0.05917498  NA   NA
```

Next, we will consider illustrating the previous set of results using the `lattice` package. The power is illustrated in Figure 1. A revised version of this illustration limiting the presentation to n =100 and displaying the type I error rate α is shown in Figure 2.

4 Example 3

We can also use the `xtable` package to summarize the results in a table. For this illustration, we consider a subset of the rows from Example 2.

```
cols<-c("n","GHR","erate","raf","pow0")
res3<-subset(res2,GHR==1.25&raf==0.3&n==500,select=cols)
res3
```

```
print(xtable(res3,digits=c(0,0,1,1,1,3)),
      include.rownames=FALSE,floating=FALSE)
```

n	GHR	erate	raf	pow0
500	1.2	0.5	0.3	0.641
500	1.2	0.7	0.3	0.769
500	1.2	0.9	0.3	0.852

Table 1: Tabular summary of the results from Example 2

```
##              n  GHR erate raf      pow0
## power314   500 1.25   0.5 0.3 0.6407236
## power1114 500 1.25   0.7 0.3 0.7694030
## power2114 500 1.25   0.9 0.3 0.8515087
```

The corresponding table generated by `xtable` is shown in Table~1.

5 Miscellaneous

The `tikzDevice` and `latticeExtra` packages can be used to considerably refine the illustrations. The software development repository provides some examples `bitbucket.org/kowzar/survsnp/`.

The session information is provided in Table~2

```
KEY=paste("q=",levels(factor(res2$raf)),sep="")
KEY<-list(lines=list(col=1:length(KEY),lty=1:length(KEY)),
          text=list(labels=paste("q=",levels(factor(res2$raf)),sep="")),
          column=3)
print(xyplot(pow0~GHR|factor(erate)*factor(n),group=factor(raf),
             data=res2,type="l",lty=KEY$lines$lty,col=KEY$lines$col,
             key=KEY,
             xlab="Genotype Hazard Ratio",ylab="Power"))
```

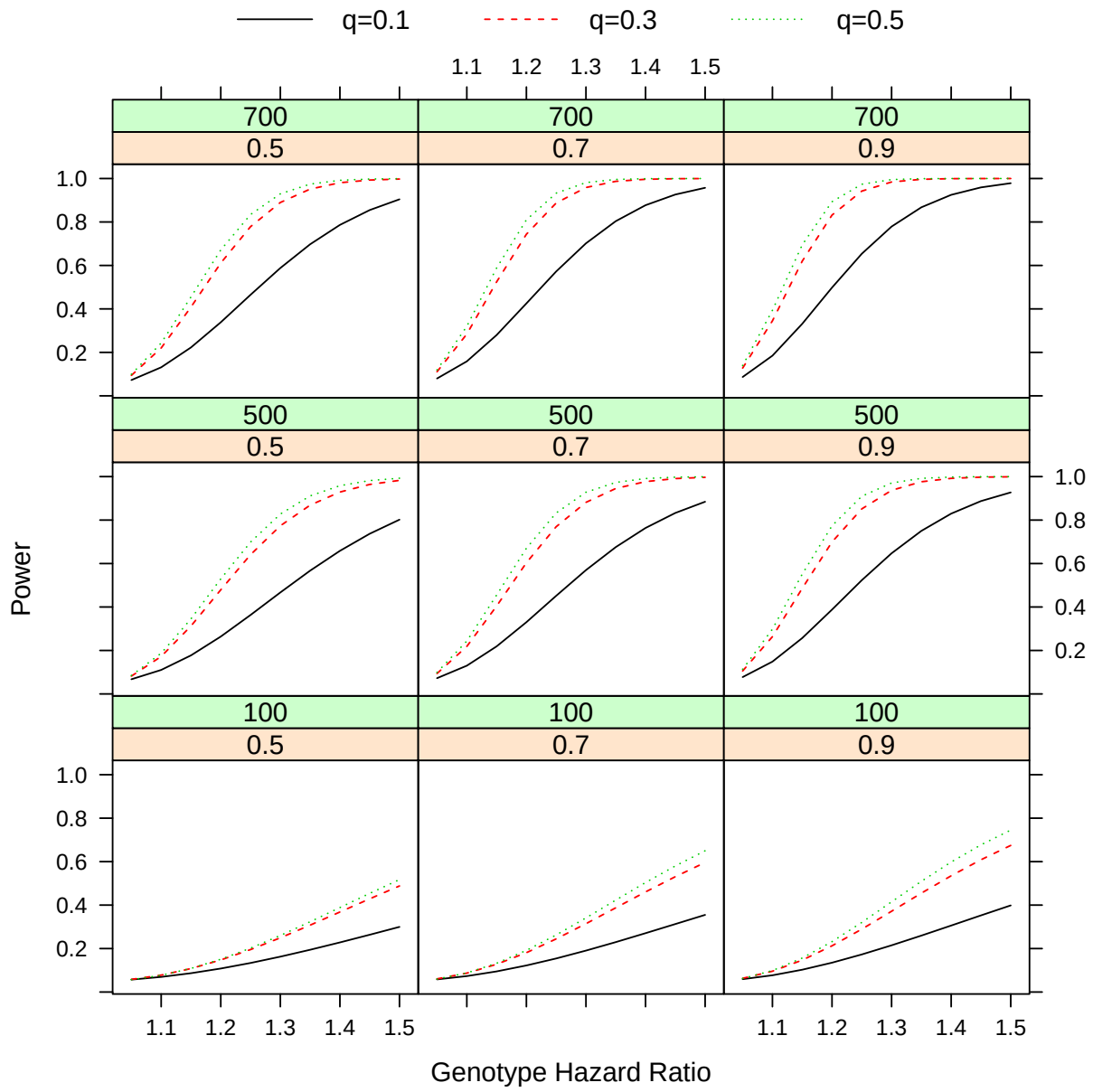

Figure 1: Power Illustration for Example 1.

```
print(xyplot(pow0~GHR|factor(erate),group=factor(raf),
             data=subset(res2,n==ns[1]),
             type="l",lty=KEY$lines$lty,col=KEY$lines$col,
             key=KEY,
             xlab="Genotype Hazard Ratio",ylab="Power",
             sub=paste("n=",ns[1],", alpha=",round(unique(res2$alpha),2))))
```

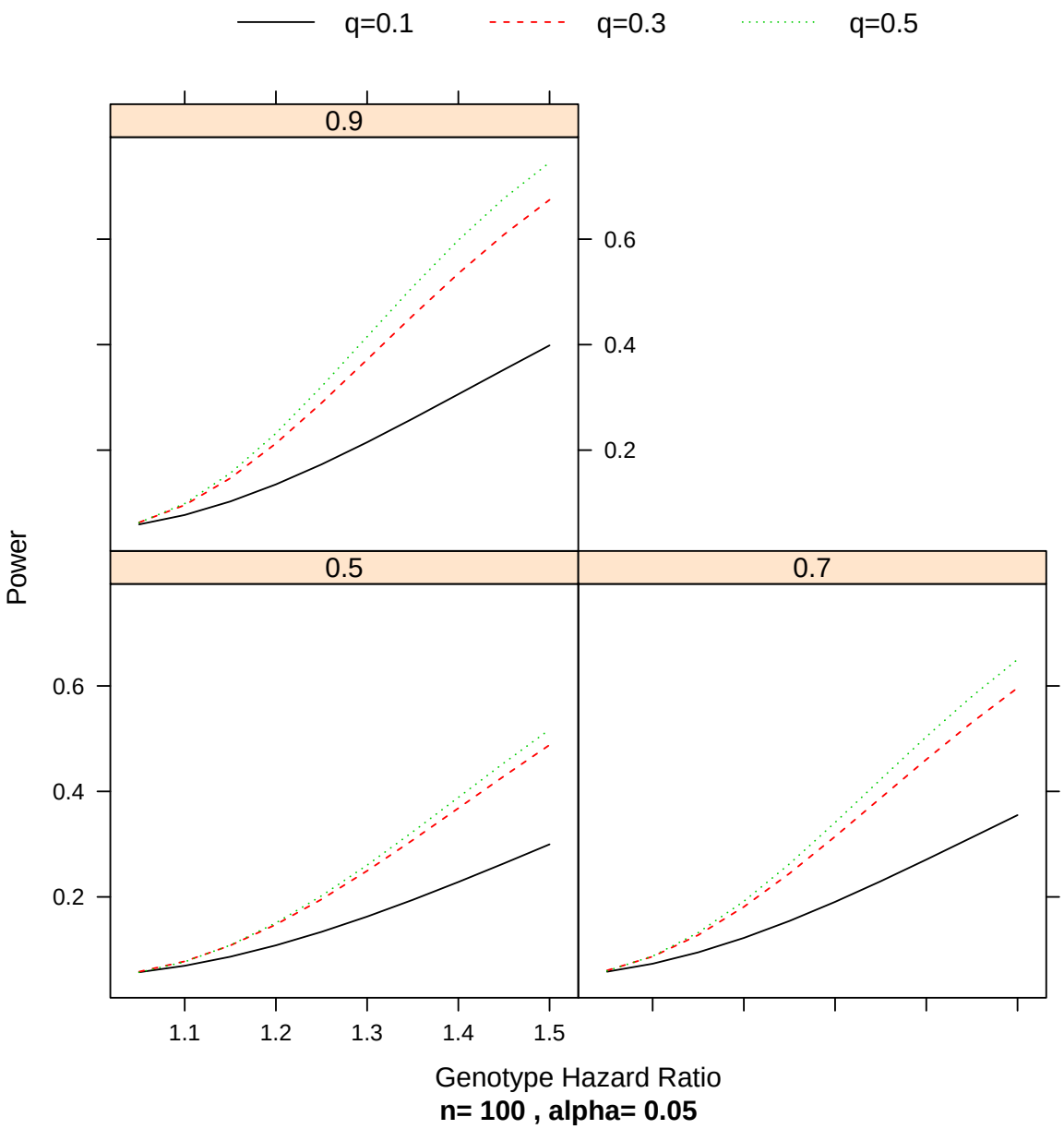

Figure 2: Power Illustration for Example 1 (restricted to n =100).

- R version 3.1.2 (2014-10-31), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, splines, stats, utils
- Other packages: Rcpp~0.11.3, foreach~1.4.2, lattice~0.20-29, survSNP~0.23.2, survival~2.37-7, xtable~1.7-4
- Loaded via a namespace (and not attached): codetools~0.2-9, compiler~3.1.2, evaluate~0.5.5, formatR~1.0, grid~3.1.2, highr~0.4, iterators~1.0.7, knitr~1.8, stringr~0.6.2, tools~3.1.2

Table 2: Session Information