Tutorial 9 - Bootstrap Methods (Solutions).

Christopher A Oldnall

Welcome to the ninth tutorial of the Regression and Simulation methods module. This is the next script in developing your skills in R, whilst learning about bootstrap methods. The questions here are a mix of newly written questions which function to be similar enough to the assignment as to provide the skills you need but not similar enough to just allow you to get the answers.

Exercise 0

Throughout remember we will need tidyverse, go ahead and do this as your first task.

Your Answer:

```
library(tidyverse)
```

Exercise 1: Non-Parametric Bootstrap

Given the following data sets, run a bootstrap with 1000 samples to derive a 95% CI for both the mean of X and for the mean of Y.

```
x<-c(0.4,0.7,1.1,0.2,0.7,1.4,1.3,0.2)
y<-c(22,5,11,9,9,5,2,7)
meansX<-rep(NA,5000)
meansY<-rep(NA, 5000)
for (k in 1:5000){
  yboot<-sample(y,8,replace=TRUE)
  xboot<-sample(x,8,replace=TRUE)
  ymeanboot<-mean(yboot)
  meansY[k]<-ymeanboot
  xmeanboot<-mean(xboot)
  meansX[k]<-xmeanboot
}
sort(meansX)[c(125,4875)]
```

```
## [1] 0.4500 1.0625
```

```
sort(meansY)[c(125,4875)]
```

```
## [1]  5.375 13.000
```

Exercise 2

Fit a Poisson GLM to the data X and Y. Interpret your findings.

```
mod<-glm(y~x,family=poisson)
summary(mod)
```

```
##
## Call:
## glm(formula = y ~ x, family = poisson)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.6350     0.2133  12.353   <2e-16 ***
## x            -0.6807     0.2843  -2.394   0.0167 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 26.396  on 7  degrees of freedom
## Residual deviance: 20.384  on 6  degrees of freedom
## AIC: 55.061
##
## Number of Fisher Scoring iterations: 4
```

Exercise 3

Calculate the CI of the effect estimate of x on y via the approximate distribution given by the GLM.

```
mod$coeff[2]+qnorm(c(0.025,0.975))*0.2843
```

```
## [1] -1.2378972 -0.1234616
```

Exercise 4

Calculate the CI via a parametric bootstrap using 5000 re-samples.

```
betas<-rep(NA,5000)
for (k in 1:5000){
  yboot<-rpois(8,exp(mod$coeff[1]+mod$coeff[2]*x))
  modboot<-glm(yboot~x,family=poisson)
  betas[k]<-modboot$coeff[2]
}
sort(betas)[c(125,4875)]
```

```
## [1] -1.2851712 -0.1365119
```

Exercise 5

Calculate the CI via the PITR bootstrap using 5000 re-samples

```r
PITRbetas<-rep(NA,4999)
for (k in 1:4999){
  us<-ppois(y,exp(mod$coeff[1]+mod$coeff[2]*x))
  us<-sample(us,size=8,replace=TRUE)
  yboot<-qpois(us,exp(mod$coeff[1]+mod$coeff[2]*x))
  modboot<-glm(yboot~x,family=poisson)
  PITRbetas[k]<-modboot$coeff[2]
}
sort(betas)[c(125,4875)]
```

```
## [1] -1.2851712 -0.1365119
```

Exercise 6

Compare the CIs from 3.1-3.3. What are the differences? Which seems to be the most conservative. What other bootstrap method could you employ to solve problems mentioned?

The difference is caused by overdispersion, which makes both 3.1 and 3.2 inadequate (since those methods are based on the assumption that the fitted model is adequate). You could employ a non-parametric bootstrap method here and that would also solve this issue.
