R Camp 2019 HW4

Alex Pugh

8/6/2019

Problem 1

A leader's approval rating is thought to represent a leader's political capital; popular leaders are more likely to be able to get their preferred policies passed than less popular leaders. There are many factors that likely determine a leader's approval rating. One potential factor that I am interested in the whether a leader entered office with a different source of support than the previous leader. Leaders who enter office with a different source of support than the previous leader may be more likely to enjoy higher initial approval ratings because they come to office with a cleaner slate. Leaders who follow another leader of the same source of support will likely be tied to the policies of the previous leader, resulting in some outside the source of support initally having a more negative view of the new leader, even though they have not made any policies.

To explore this, I will consider the average two-week approval rating of leaders in the US. I expect that being a leader with a new source of leader support (SOLS) will have a positive effect on a leader's average two-week approval rating. There are other variables that will also influence a leader's initial approval rating. First, the percentage of the vote that they won likely has a strong impact on a leader's inital approval rating. Leader's who won a greater percentage of votes are likely to have higher inital approval ratings as more people supported them in the election, thus presumably approve of the leader. Another factor that I will consider is the state of the economy. While the new leader has not made any policy influencing the economy, the state of the economy is important to individuals' lives. The economy likely influences whether people vote and for whom they vote.

We should first consider the theoretical range of the average first two-week approval rating of new leaders. Approval ratings can range from 0-100%. The distribution of the average first two-week approval rating of new leaders likely follows a normal distribution with a mean around 50 or 60; extreme values are less likely while values around the mean are more likely.

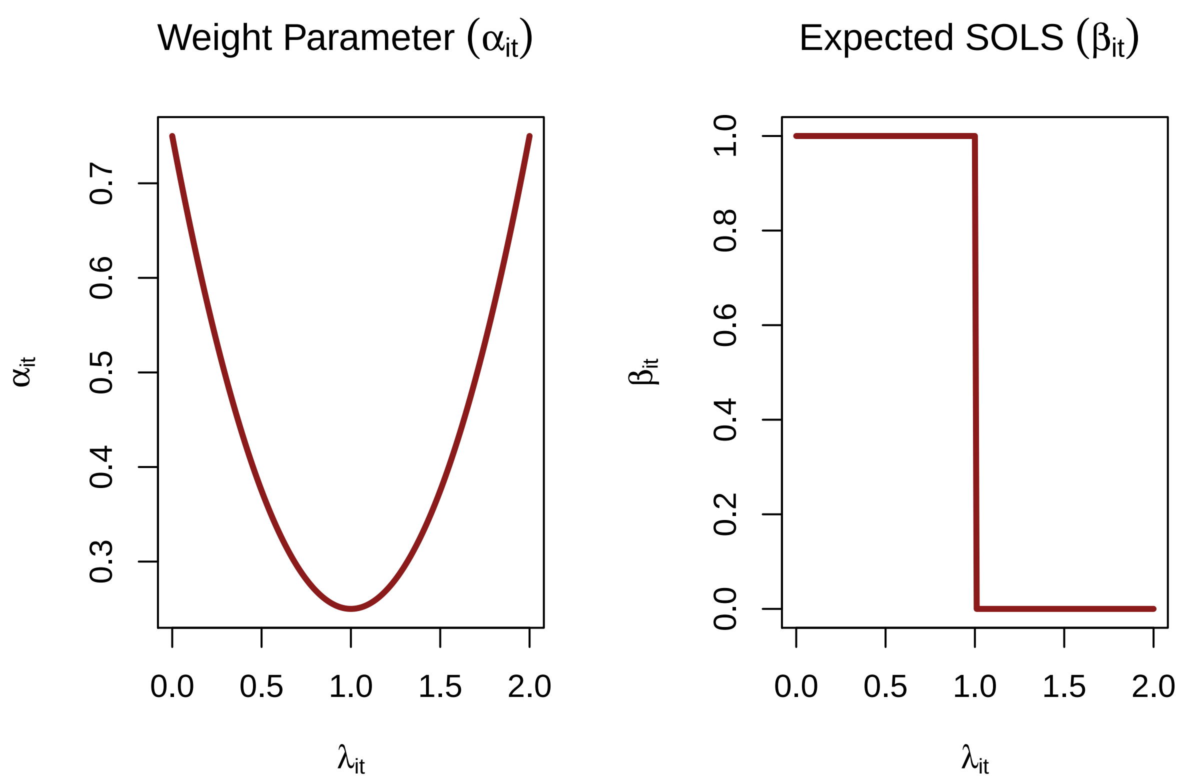
If we simplify the electoral system to a two party system in which a leader must win the majority of the vote to win, the percentage of votes that the elected leader can win ranges from 51% to 100%. Percentages closer to 51% are generally more likely than percentages closer to 100%. In a two party democratic system, we can consider each party to be a different source of leader support. If a leader is from the same party as his predecessor, there is not a SOLS change. Conversely, if a leader is from the other party than his predecessor, a SOLS change has occurred. Thus, SOLS changes either occur or not, and are coded as a dummy variable. On surveys, the percieved economy has be represented as a categorical variable taking on 3 values of poor, average, or good. We can calculate the average perception of US survey respondents, resulting in a value ranging from 0 to 2. We consider that SOLS changes are a function of the economic conditions; if the economy is good, voters do not vote out the current party but if the economy is bad, they vote for the other party. If the economy is a driver of people voting, we should expect greater percentages of the population voting when the economy is bad/good and less when the economy is average as there is less of an incentive for voters to reward or punish. In our model, we will assume that the perception of the economy has no independent impact on approval rating aside from impacting whether a SOLS change occurred and how much of the population turns out to vote. We expect that the impact of whether a SOLS occured and the percentage won on the approval rating are weighted by the amount of the population that voted in the election. As more people in the population vote, their impact is stronger on the national approval rating.

Formally, Pred_App = alpha_{it}* delta_{it} + alpha_{it}*beta_{it}, where alpha = weight parameter (0,1) (how much of the population voted in election) delta = percentage won (.51,1.00) beta = expected SOLS change based on economy (0, 1)
lambda = averge perception of the economy

alpha = 2*(lambda)ˆ2-2*lambda+0.75 (within our range of lambda ranges from 0.75 to 0.25) beta = 1 - round(lambda/2) (If average perception greater than or equal to 1, no SOLS change)

We can plot the values of alpha and beta across the values of alpha to ensure that all of the reulting values make theorteical sense. Our alpha (% of population voting) ranges from 0.75 to 0.25, which cover reasonable values of the population to turn out to vote. As we expected, when lambda is at extreme values, the turnout is high, but when lambda is in the middle of its range, turnout is low. Beta (SOLS changes) can only take on values of 0 or 1. When lambda is low, a SOLS change occured, but when lambda is high, a SOLS change did not occur. Alpha and beta take on reasonable values within out theoretical ranges across the theoretical bounds of lambda

```
# Showing the behavior of parameters
  lambda = seq(from=0,to=2,by=0.01)     # averge perception of the economy
  alpha = 0.5*(lambda^2)-lambda+0.75           # Weight parameter
  beta = 1 - round(lambda/2)              # SOLS Change occurring



   par(mfrow=c(1,2))
   plot(alpha ~ lambda, type="l", col="firebrick4", lwd=3,
        xlab=expression(lambda[it]), ylab=expression(alpha[it]),
        main=expression(paste("Weight Parameter ", (alpha[it]), sep=" ")))
   plot(beta ~ lambda, type="l",  col="firebrick4", lwd=3,
        xlab=expression(lambda[it]), ylab=expression(beta[it]), ylim=c(0,1),
        main=expression(paste("Expected SOLS ", (beta[it]), sep=" ")))
```

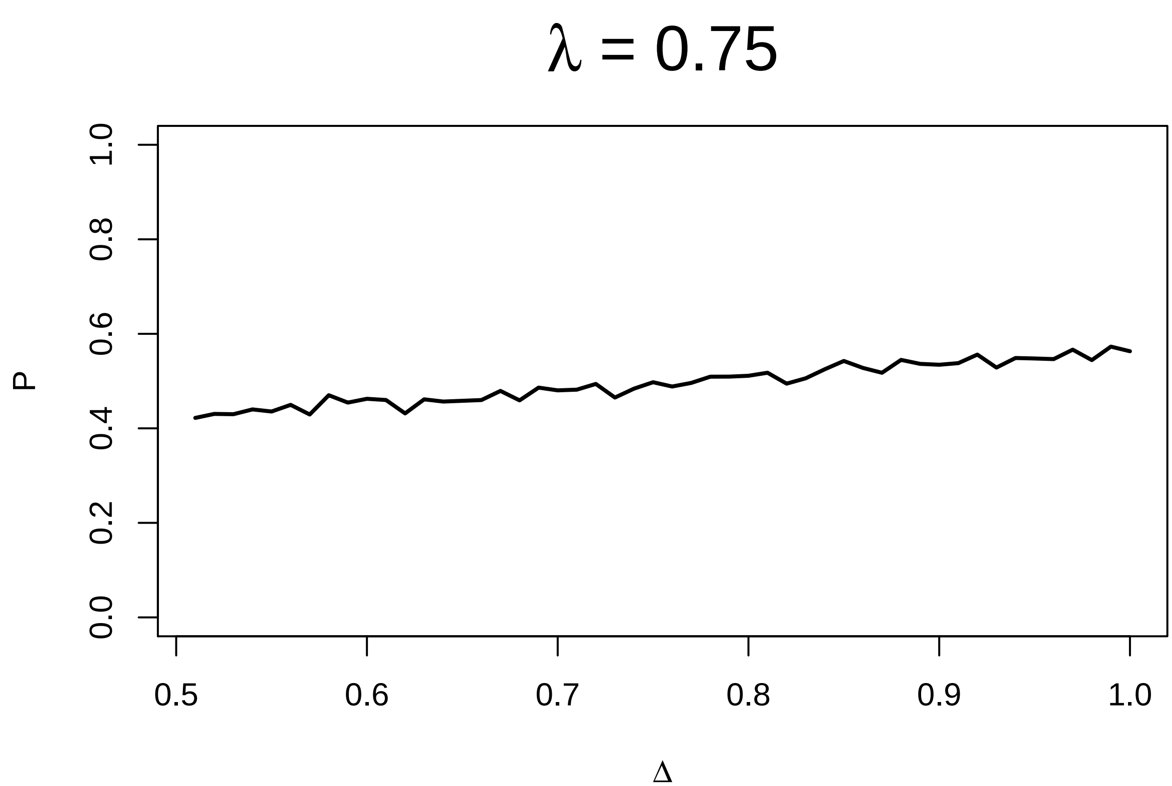


We can also explore the output of the model by setting lambda (average economic perception) at a value and plotting across the range of our delta (percentage of votes won). If we set lambda at 0.75, indicating a generally poor averge perception of the economy, the plot below show the early term approval rating of a new leader.

```
# Exploring the functional form of the model
rm(list=ls())
par(mfrow=c(1,1))

#set alpha value
lambda_val = 0.75   # Tuning parameter (economic perception)

delta = seq(from=.51, to=1, by=0.01)
alpha = 0.5*(lambda_val^2)-lambda_val+0.75          # Weight parameter
beta = 1 - round(lambda_val/2)
error = rnorm(length(delta), mean = 0, sd = 0.01)
q = alpha * delta + alpha*beta + error
summary(q)
```

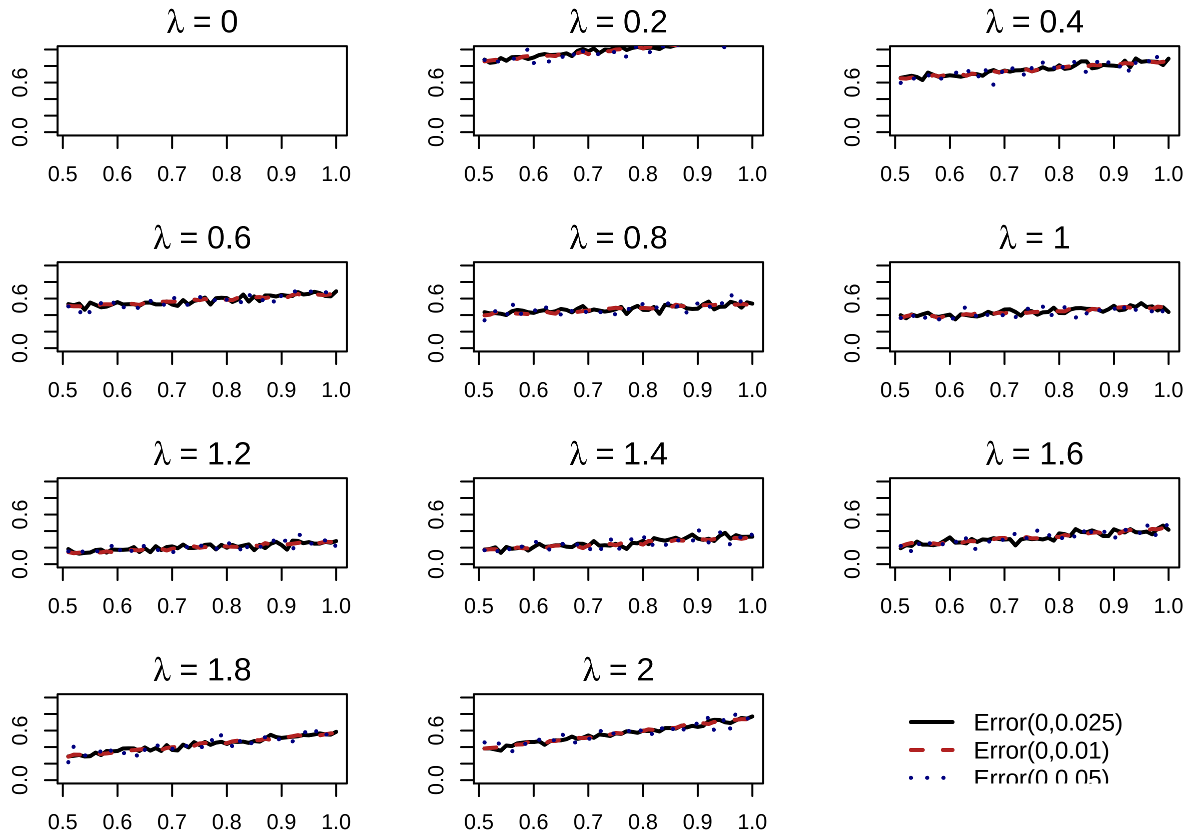
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4221  0.4599  0.4943  0.4951  0.5331  0.5729
```

```
# Sample plot (We want to create a bunch of these by varying parameters)
plot(q ~ delta, type="l", lwd=2, ylab="P", ylim=c(0,1), cex.main=2, xlab=expression(Delta),
     main=substitute(paste(lambda, sep=" = ", v), list(v=lambda_val)))
```



Instead of setting the value of lambda to one value, we can combine multiple graphs into one plot that shows how the approval rating changes across different values of lambda. Within each graph, we can adjust the amount of error introduced into the model by changing the standard deviation. We assume that SOLS changes occur when most people think that the economy is bad. We can see that the inital approval rates of a new leader are high when the perception of the economy is lower, indicating a SOLS change has occurred.

```
  rm(list=ls())
#create space to plot graphs
  par(mfrow=c(4,3),  mar = c(3, 3, 2, 2))
#set seed
  set.seed(2418)
#create vector for lambda
lambda <- seq(from = 0, to = 20, by = 2)

  # Write a loop for this
  for(i in lambda){                                 # We explore (0,2) in lambda

    lambda_val = i * 0.1                          # Parameters to be varied
    SD = 0.025                                       # Parameters to be varied
    SD_small = 0.01                                  # Parameters to be varied
    SD_large = 0.05                                 # Parameters to be varied

   delta = seq(from=.51, to=1, by=0.01)
   alpha = 0.5*(lambda_val^2)-lambda_val+0.75
   beta = 1 - round(lambda_val/2)

    #Change standard deviation in error term
    error = rnorm(length(delta), mean = 0, sd = SD)
    pred1  <- alpha * delta + alpha*beta + error
    error = rnorm(length(delta), mean = 0, sd = SD_small)
    pred2 <- alpha * delta + alpha*beta + error
    error = rnorm(length(delta), mean = 0, sd = SD_large)
    pred3 <- alpha * delta + alpha*beta + error

    plot(pred1 ~ delta, type="l", lwd=2, ylim=c(0,1), ylab="",
        xlab=expression(Delta), cex.main=1.5,
        main=substitute(paste(lambda, sep=" = ", v), list(v=lambda_val)))
    lines(pred2 ~ delta, type="l", lty=2, col="firebrick", lwd=2)
    lines(pred3 ~ delta, type="l", lty=3, col="navy", lwd=2)
     }

  # We want to put a legend on this combined graph
  plot(1, type = "n", axes=FALSE, xlab="", ylab="") # No plotting
  legend(x = "topleft",
         legend = c("Error(0,0.025)", "Error(0,0.01)", "Error(0,0.05)"),
         col=c("black", "firebrick", "navy"),
         lty=c(1,2,3),
         lwd=2, cex=1.1, horiz = FALSE, text.width=0.1, box.col = "white")
```

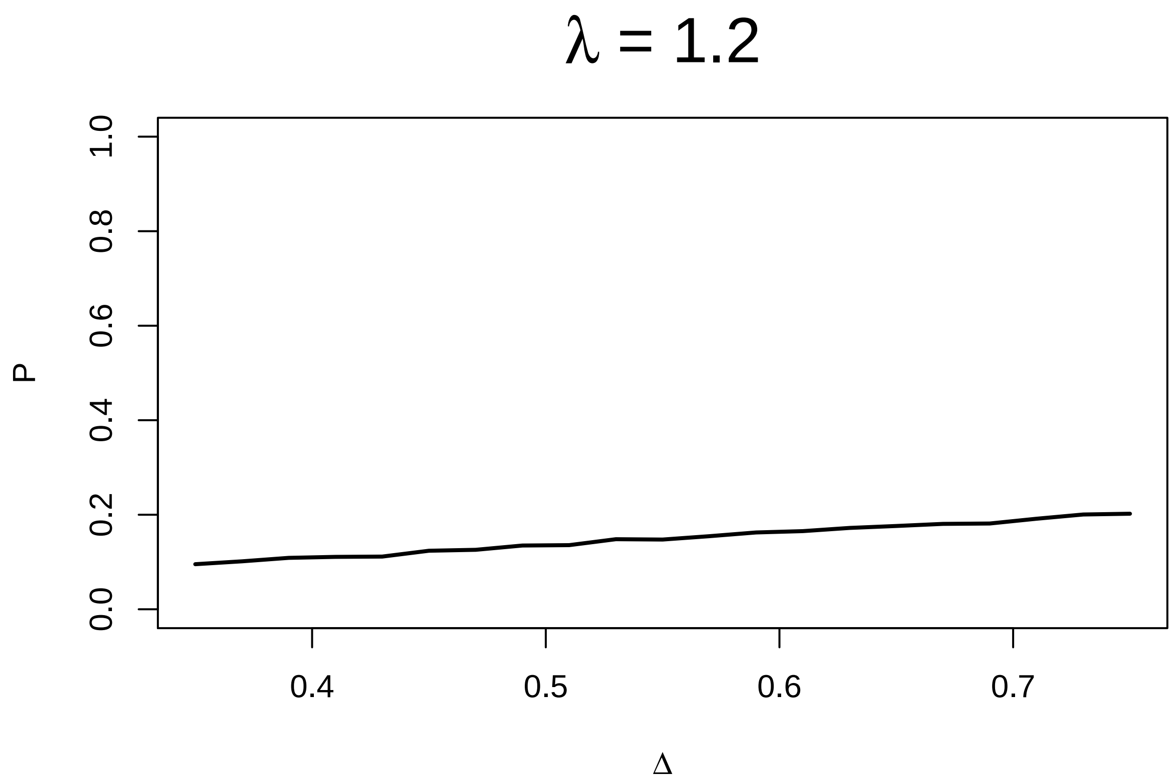


Problem 2

Now we can write a package that will allow the user to set the value of lambda, the range of delta to plot over, and the standard deviation of the error term in the model.

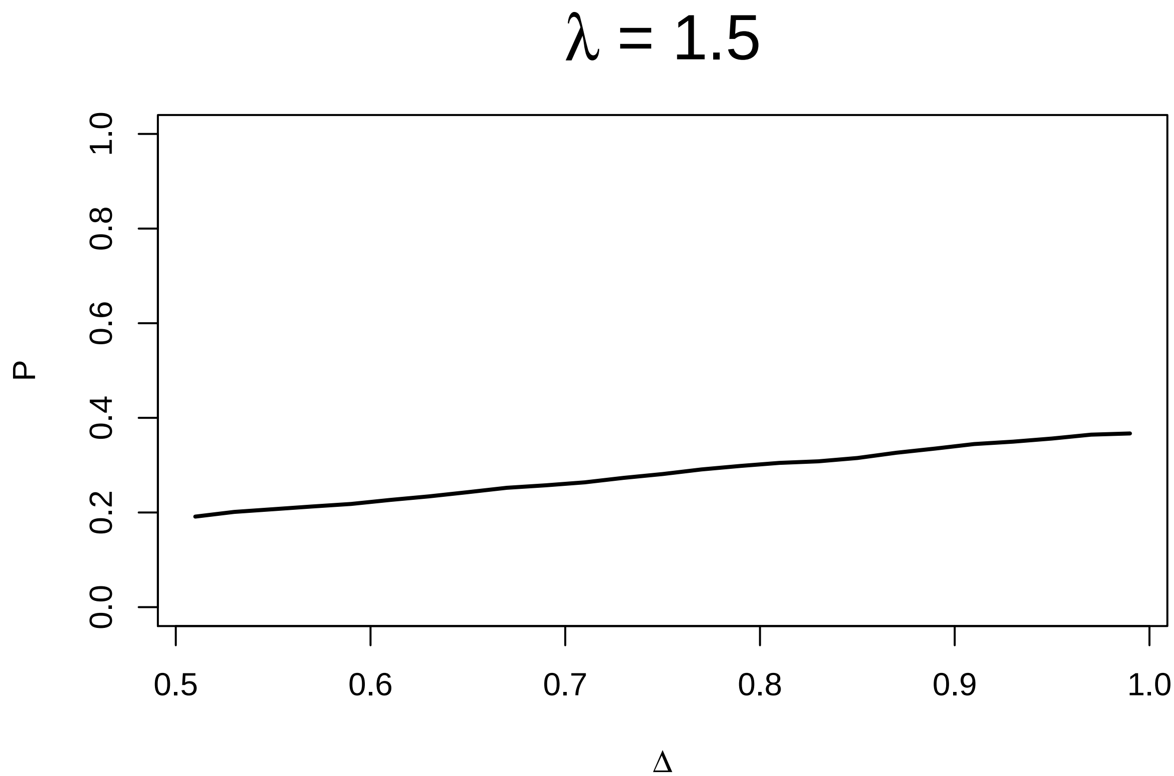
```
PlotParm <- function(lvalue, dstart, dend, dby, eSD){
    par(mfrow=c(1,1))

   #set alpha value
   lambda_val = lvalue    # Tuning parameter (economic perception)

   delta <- seq(from=dstart, to=dend, by=dby)
   alpha = 0.5*(lambda_val^2)-lambda_val+0.75          # Weight parameter
   beta = 1 - round(lambda_val/2)
   error = rnorm(length(delta), mean = 0, sd = eSD)
   q = alpha * delta + alpha*beta + error

   # Sample plot (We want to create a bunch of these by varying parameters)
   plot(q ~ delta, type="l", lwd=2, ylab="P", ylim=c(0,1), cex.main=2, xlab=expression(Delta),
        main=substitute(paste(lambda, sep=" = ", v), list(v=lambda_val)))

   return(summary(q))
}
```

```r
#demonstrate function works
PlotParm(1.2, 0.35, 0.75, 0.02, 0.002)
```



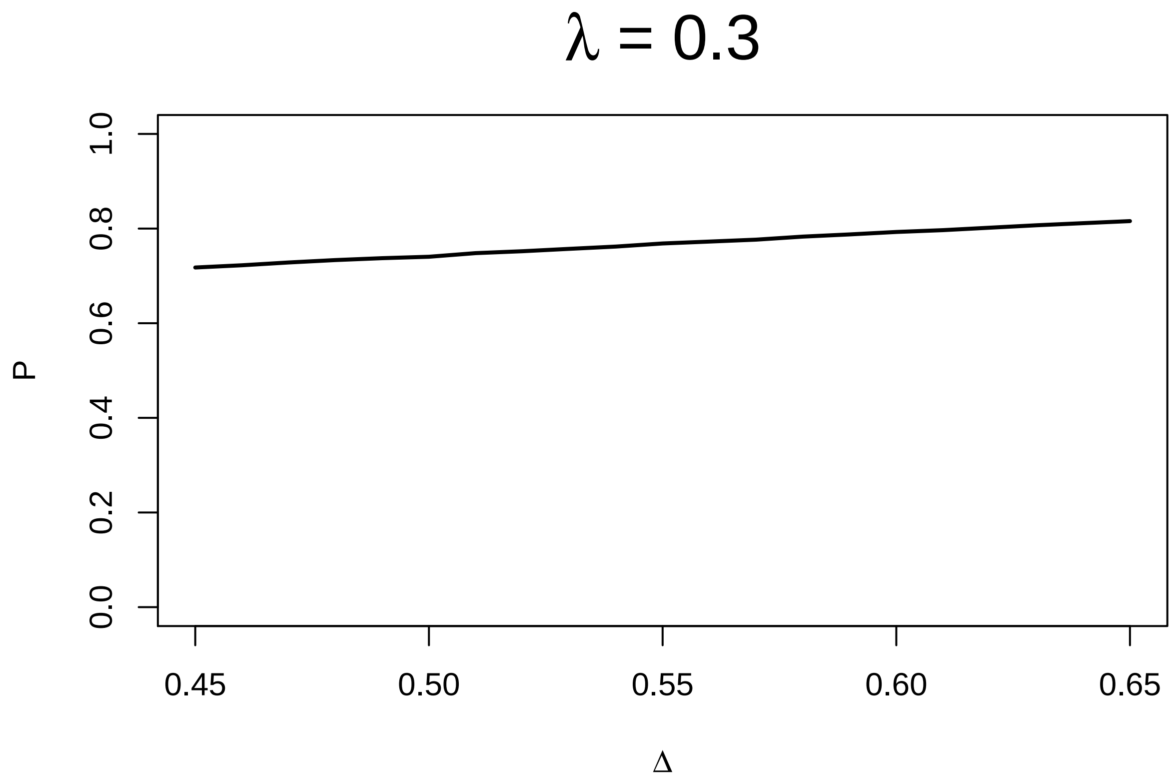

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.09534 0.12381 0.14825 0.14908 0.17621 0.20212
```

```r
PlotParm(1.5, 0.51, 0.99, 0.02, 0.002)
```



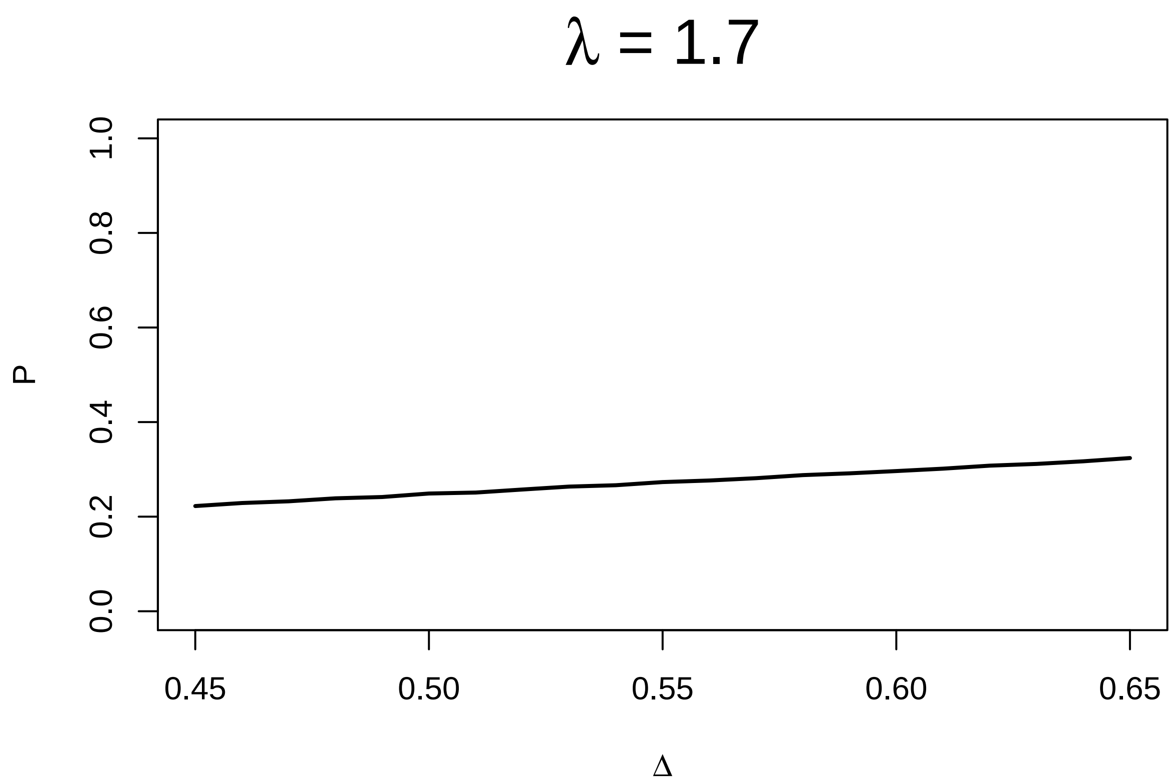

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1913  0.2341  0.2812  0.2809  0.3263  0.3670
```

```
PlotParm(0.3, 0.45, 0.65, 0.01, 0.001)
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.7177  0.7405  0.7685  0.7673  0.7928  0.8158
```

```
PlotParm(1.7, 0.45, 0.65, 0.01, 0.001)
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2224  0.2489  0.2731  0.2724  0.2964  0.3240
```