Examples

We begin by demonstrating the `find_eps` function on Example 1 from Section 4 and Appendix C.1 In the case where $\tilde{r} = 3$, we define the function `r_star` as follows and plug the function into `find_eps`.

```
q_tilde <- 1; r_tilde <- 3; a_tilde <- 0.25

r_star <- function(p,q){
  if(q != q_tilde){
    return(Inf)
  }
  else{
    return(max(a_tilde/(p*q_tilde), r_tilde))
  }
}

find_eps(r_star)
```

```
## $eps
## [1] 1.301226
##
## $p
## [1] 0.084
##
## $q
## [1] 1
```

We compare to the closed form expression derived in the text.

```
log((r_tilde - a_tilde)/(1 - a_tilde))
```

```
## [1] 1.299283
```

We note that the closed form expression differs very slightly from the value produced by `find_eps`. We can improve the estimate by using a finer grid, as demonstrated below. (Since when $q_i \neq \tilde{q}$, no bound is enforced, we can conduct a faster search by limiting the grid to only consider $q_i = \tilde{q}$.)

```
find_eps(r_star, p_grid = seq(1e-6, 1, 1e-6), q_grid = q_tilde)
```

```
## $eps
## [1] 1.299285
##
## $p
## [1] 0.083334
##
## $q
## [1] 1
```

We see that the result produced almost exactly matches the closed form expression.

We also demonstrate the `find_eps` function on Example 2 from Section 4 and Appendix C.2. In the case where $\tilde{a} = 0.15$, we define the function `r_star` as follows and plug the function into `find_eps`.

```
p_tilde <- 0.05; r_tilde <- 3; a_tilde <- 0.15

r_star <- function(p,q){
  if(p != p_tilde){
    return(Inf)
  }
  else{
    return(max(a_tilde/(p*q), r_tilde))
  }
}

find_eps(r_star)
```

```
## $eps
## [1] 1.209838
##
## $p
## [1] 0.05
##
## $q
## [1] 1
```

We find that the result exactly matches the closed form expression derived in the text.

```
log((a_tilde*(1-p_tilde))/(p_tilde*(1-a_tilde)))
```

```
## [1] 1.209838
```