Testing

```
library(TNDsen)
o = c(1,2,3,4)*1e8
```

Testing edge cases

Delta = 0

```
k = TND_causal_bounds(o, 0)
abs(k$upper.bound - odds.ratio(o)) < 1e-10 & abs(k$lower.bound - odds.ratio(o)) < 1e-10
```

```
## [1] TRUE
```

Gamma = 1

```
k = TND_causal_bounds(o, 0.1, 1)
abs(k$upper.bound - odds.ratio(o)) < 1e-10 & abs(k$lower.bound - odds.ratio(o)) < 1e-10
```

```
## [1] TRUE
```

xi< Inf vs xi = Inf when Gamma = Inf

```
k1 = TND_causal_bounds(o, 0.1)
k2 =  TND_causal_bounds(o, 0.1, xi=2)
abs(k1$upper.bound - k2$upper.bound) < 1e-4 & abs(k1$lower.bound - k2$lower.bound) < 1e-4
```

```
## [1] TRUE
```

Testing convergence of edge cases with Confidence Interval

```
alpha = 0.95
conf.type = c('normal', 'transformed', 'quadratic')
```

Delta = 0

```
for(conf in conf.type)
{
  k = TND_causal_bounds(o, 0, alpha = alpha, conf.type = conf)
  print(abs(k$upper.bound - odds.ratio(o)) < 1e-3 & abs(k$lower.bound - odds.ratio(o)) < 1e-3)
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

Gamma = 1

```
for(conf in conf.type)
{
  k = TND_causal_bounds(o, 0.1, 1, alpha = alpha, conf.type = 'normal')
  print(abs(k$upper.bound - odds.ratio(o)) < 1e-3 & abs(k$lower.bound - odds.ratio(o)) < 1e-3)
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

xi< Inf vs xi = Inf when Gamma = Inf

```
for(conf in conf.type)
{
  k1 = TND_causal_bounds(o, 0.1)
  k2 =  TND_causal_bounds(o, 0.1, alpha = alpha, conf.type = conf)
  print(abs(k1$upper.bound - k2$upper.bound) < 1e-2 & abs(k1$lower.bound - k2$lower.bound) < 1e-2)
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

```
for(conf in conf.type)
{
  k1 = TND_causal_bounds(o, 0.1, xi=2)
  k2 = TND_causal_bounds(o, 0.1, xi=2, alpha = alpha, conf.type = conf)
  print(abs(k1$upper.bound - k2$upper.bound) < 1e-2 & abs(k1$lower.bound - k2$lower.bound) < 1e-2)
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

```
for(conf in conf.type)
{
  k1 = TND_causal_bounds(o, 0.1, alpha = alpha, conf.type = conf)
  k2 =  TND_causal_bounds(o, 0.1, xi=2, alpha = alpha, conf.type = conf)
  print(abs(k1$upper.bound - k2$upper.bound) < 1e-2 & abs(k1$lower.bound - k2$lower.bound) < 1e-2)
}
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
```