

Cost-effectiveness acceptability curve plots

```
library(BCEA)
library(dplyr)
library(reshape2)
library(ggplot2)
library(purrr)
```

The intention of this vignette is to show how to plot different styles of cost-effectiveness acceptability curves using the BCEA package.

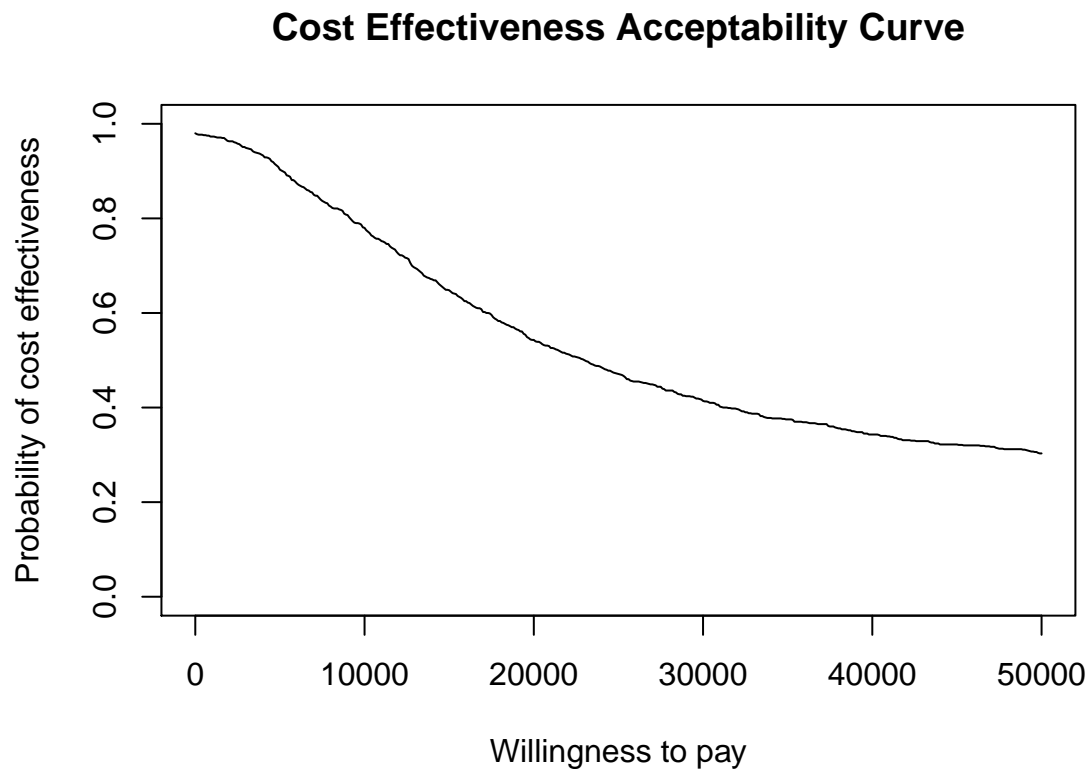
Two interventions only

This is the simplest case, usually status-quo versus an alternative intervention.

```
data("Vaccine")

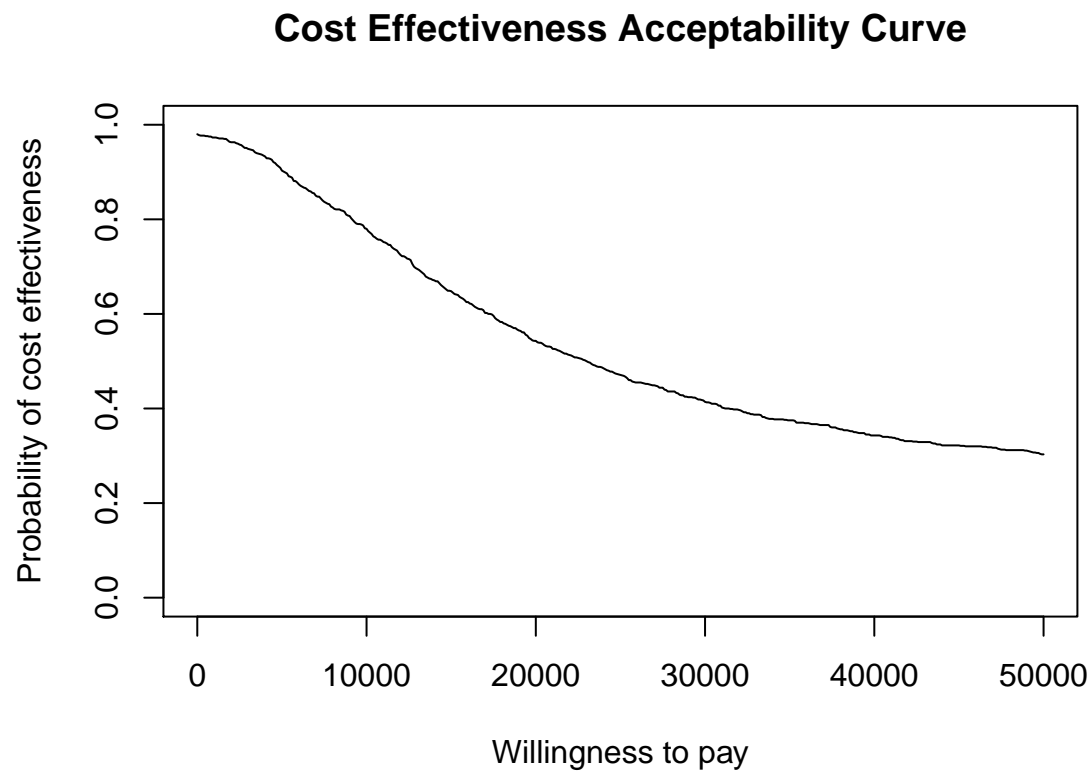
he <- bcea(e, c)
# str(he)

ceac.plot(he)
```

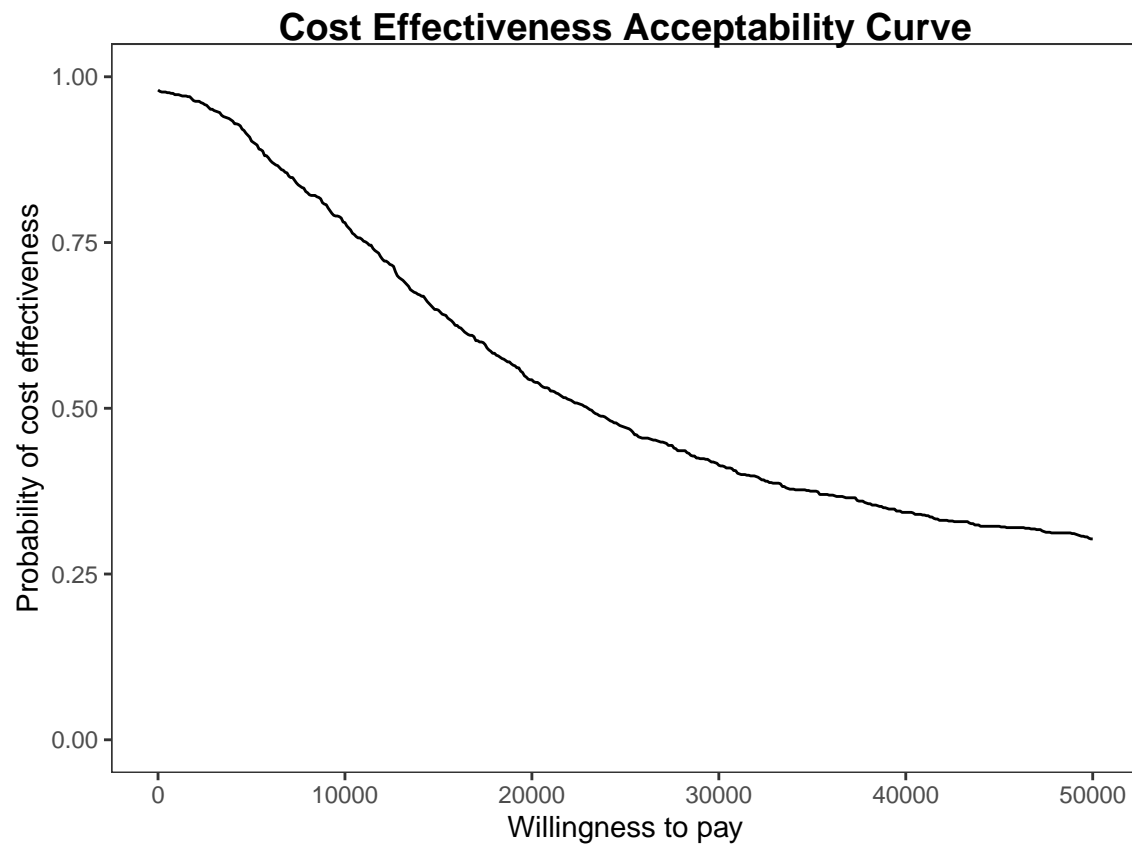


The plot defaults to base R plotting. Type of plot can be set explicitly using the `graph` argument.

```
ceac.plot(he, graph = "base")
```



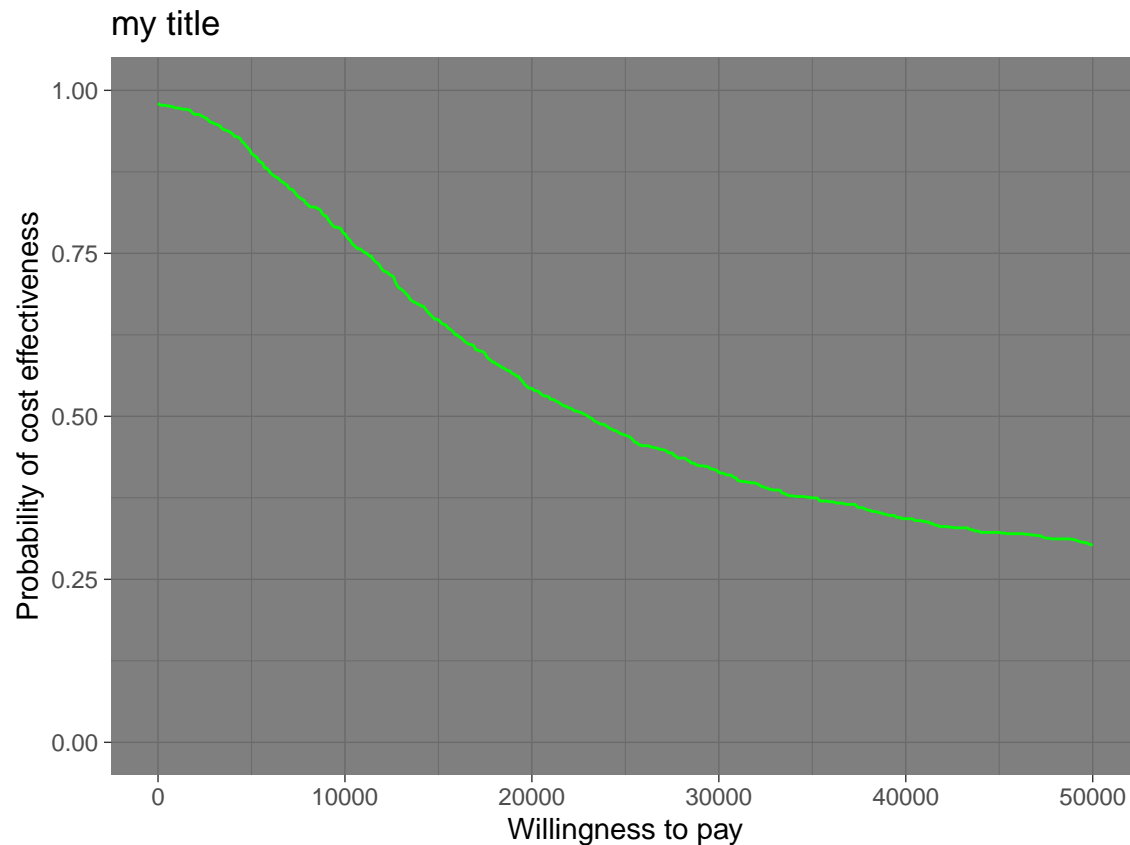
```
ceac.plot(he, graph = "ggplot2")
```



```
# ceac.plot(he, graph = "plotly")
```

Other plotting arguments can be specified such as title, line colours and theme.

```
ceac.plot(he,  
  graph = "ggplot2",  
  title = "my title",  
  line = list(colors = "green"),  
  theme = theme_dark())
```



Multiple interventions

This situation is when there are more than two interventions to consider. Incremental values can be obtained either always against a fixed reference intervention, such as status-quo, or for all pair-wise comparisons.

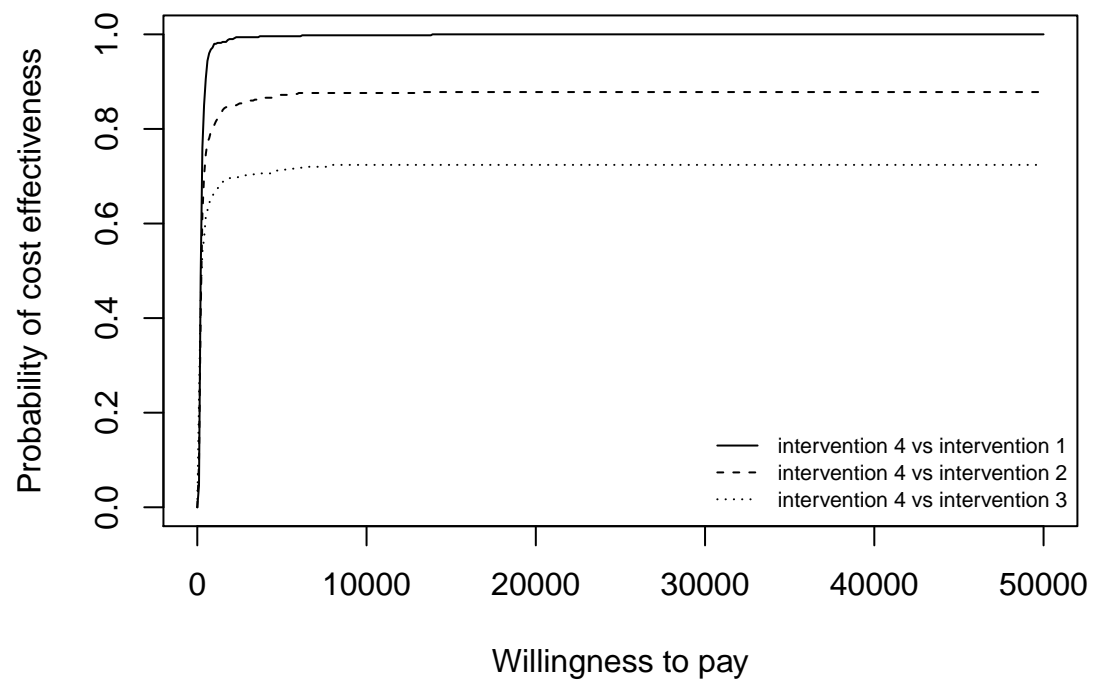
Against a fixed reference intervention

```
data("Smoking")
```

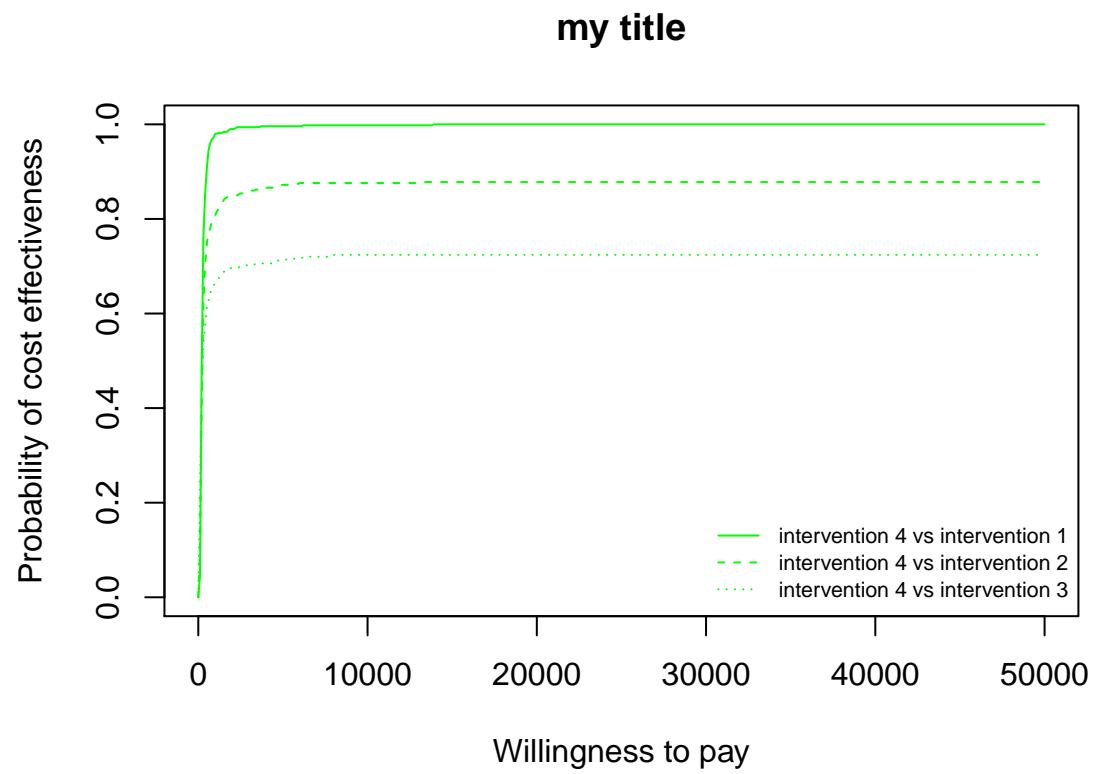
```
he <- bcea(e, c, ref = 4)  
# str(he)
```

```
ceac.plot(he)  
#> Wrong number of colours provided. Falling back to default
```

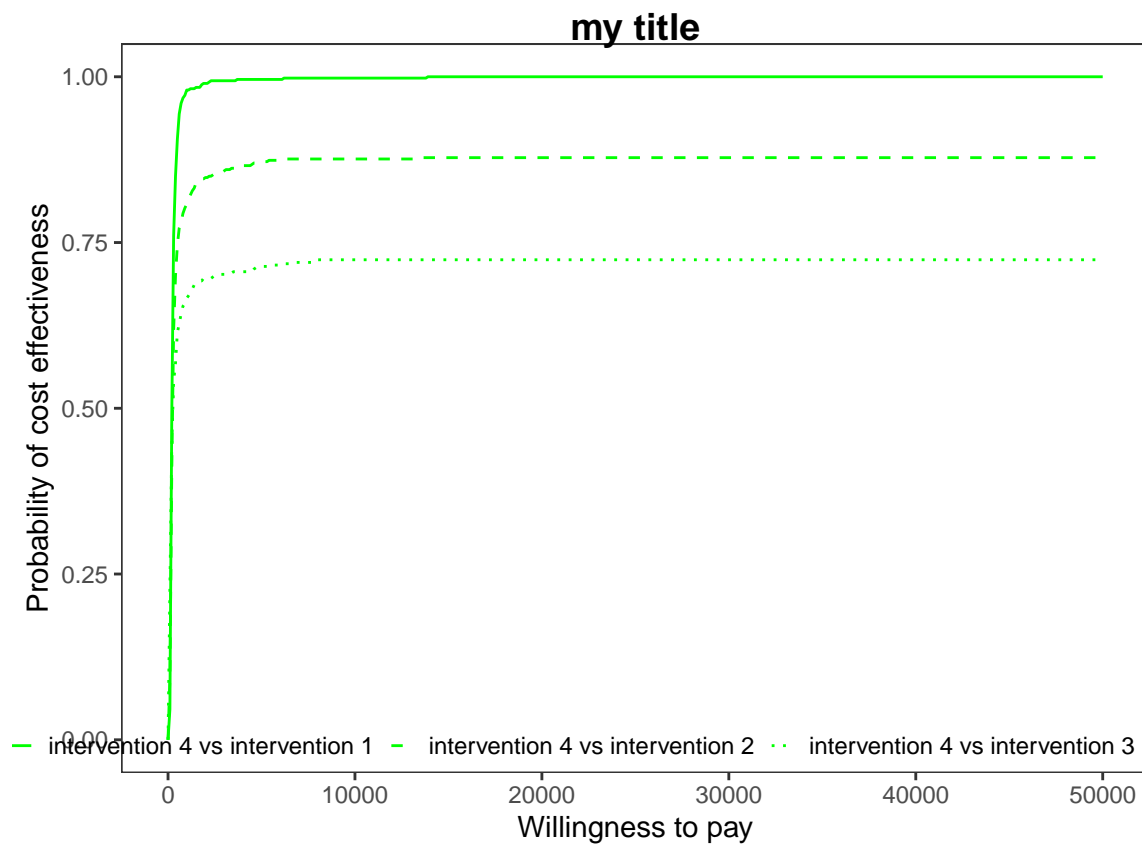
Cost Effectiveness Acceptability Curve



```
ceac.plot(he,  
  graph = "base",  
  title = "my title",  
  line = list(colors = "green"))  
#> Wrong number of colours provided. Falling back to default
```



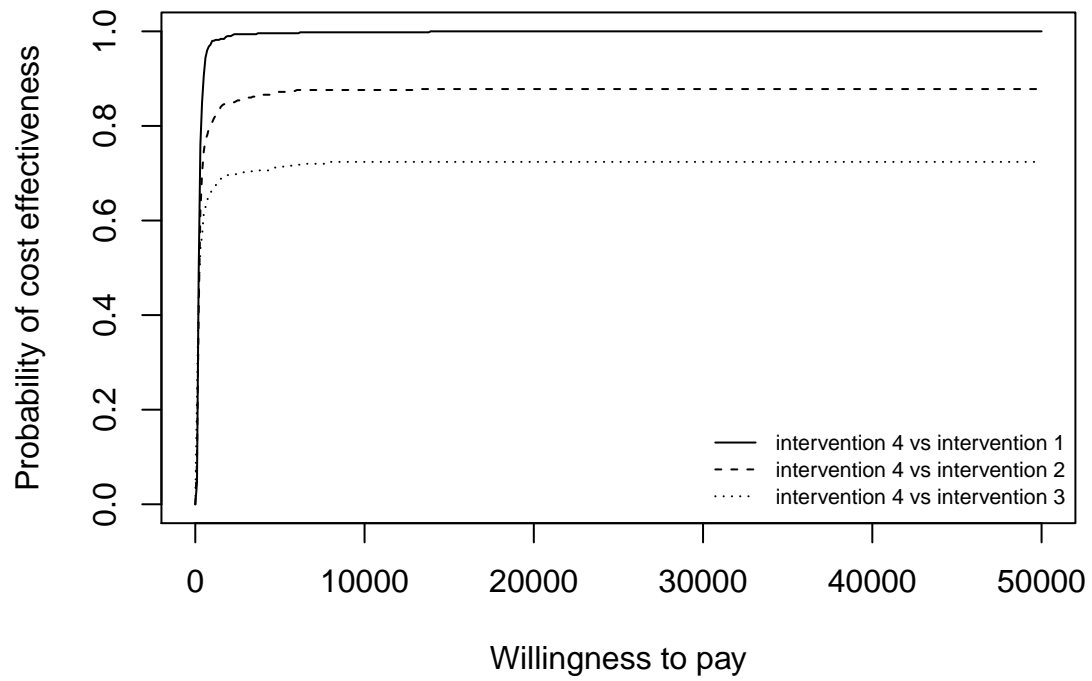
```
ceac.plot(he,  
  graph = "ggplot2",  
  title = "my title",  
  line = list(colors = "green"))
```



Reposition legend.

```
ceac.plot(he, pos = FALSE) # bottom right
#> Wrong number of colours provided. Falling back to default
```

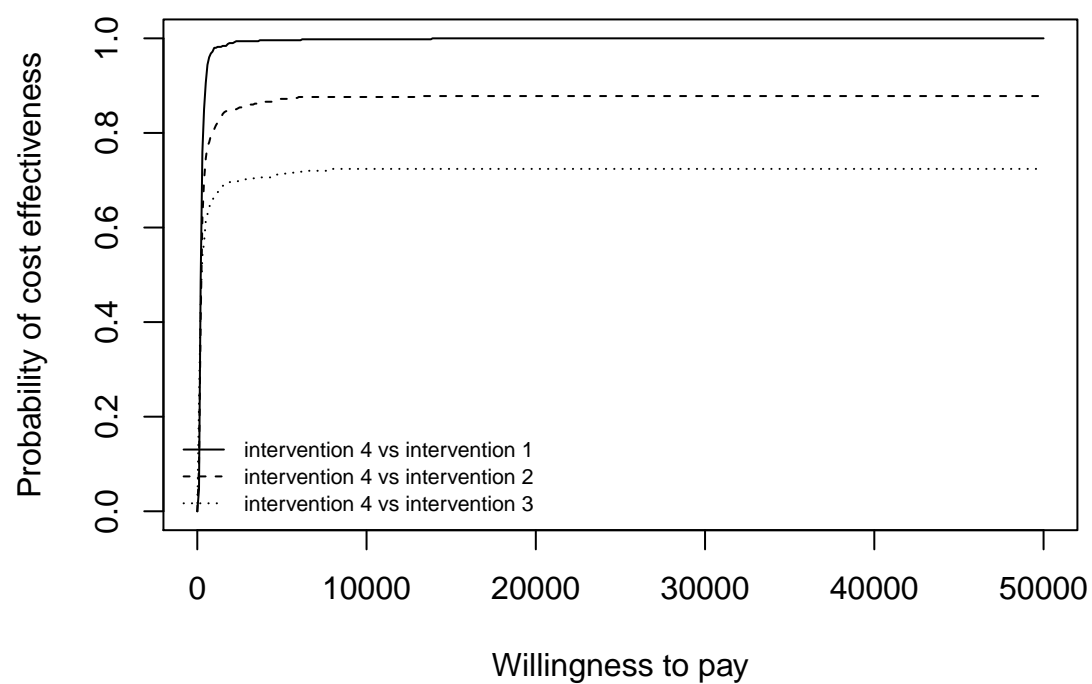
Cost Effectiveness Acceptability Curve



```
ceac.plot(he, pos = c(0, 0))
```

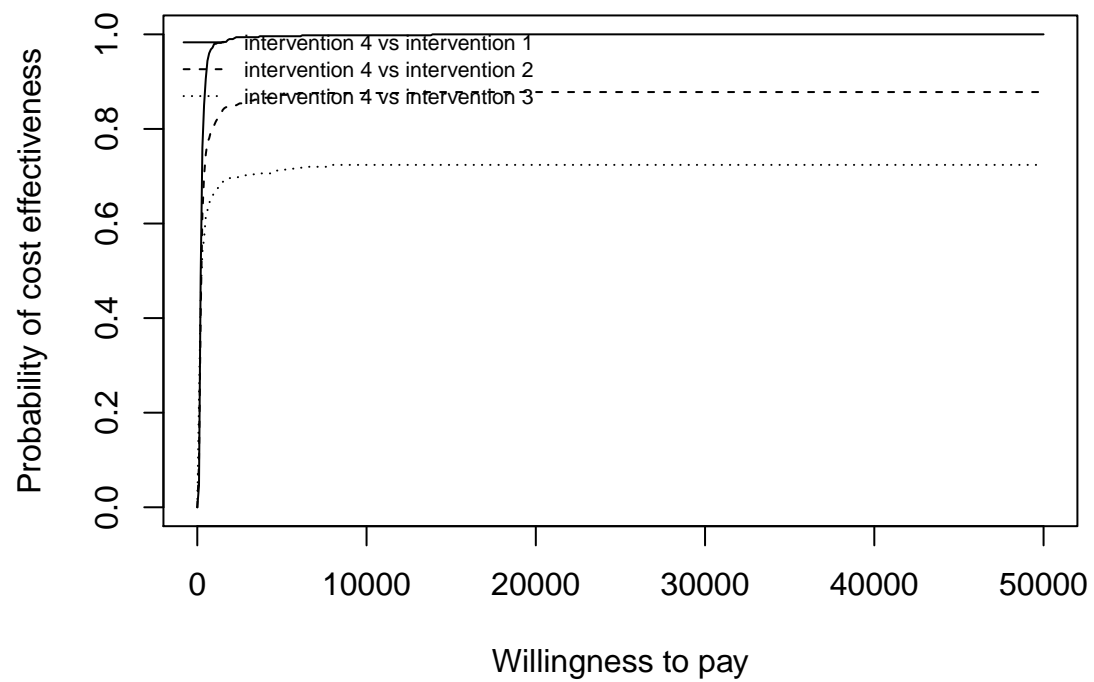
```
#> Wrong number of colours provided. Falling back to default
```


Cost Effectiveness Acceptability Curve



```
ceac.plot(he, pos = c(0, 1))  
#> Wrong number of colours provided. Falling back to default
```

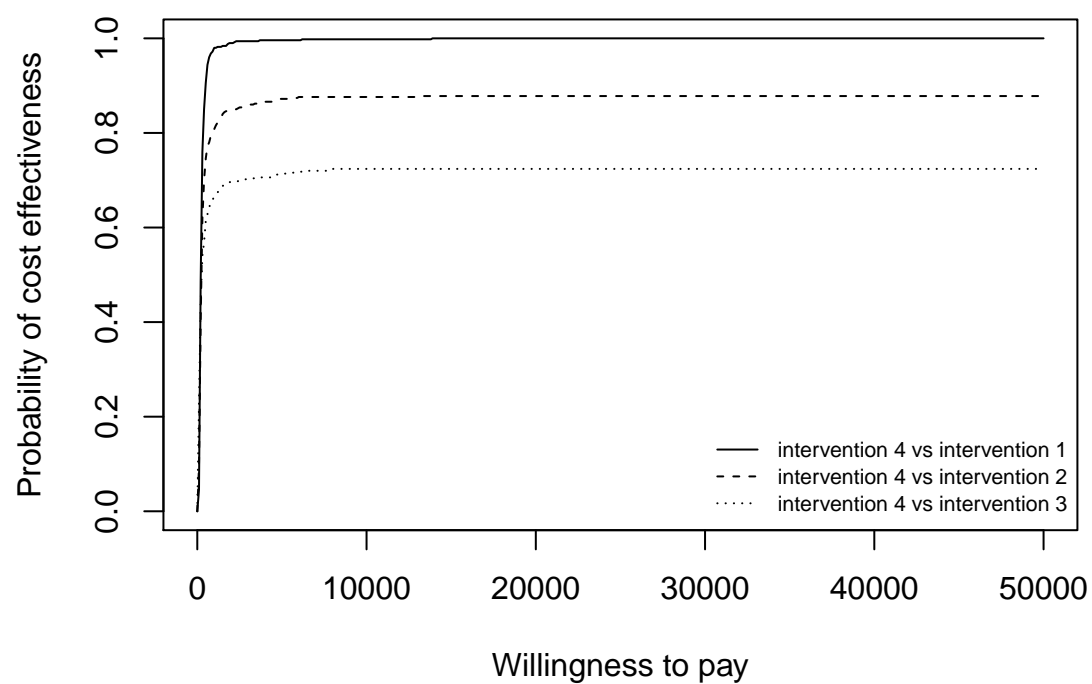
Cost Effectiveness Acceptability Curve



```
ceac.plot(he, pos = c(1, 0))
```

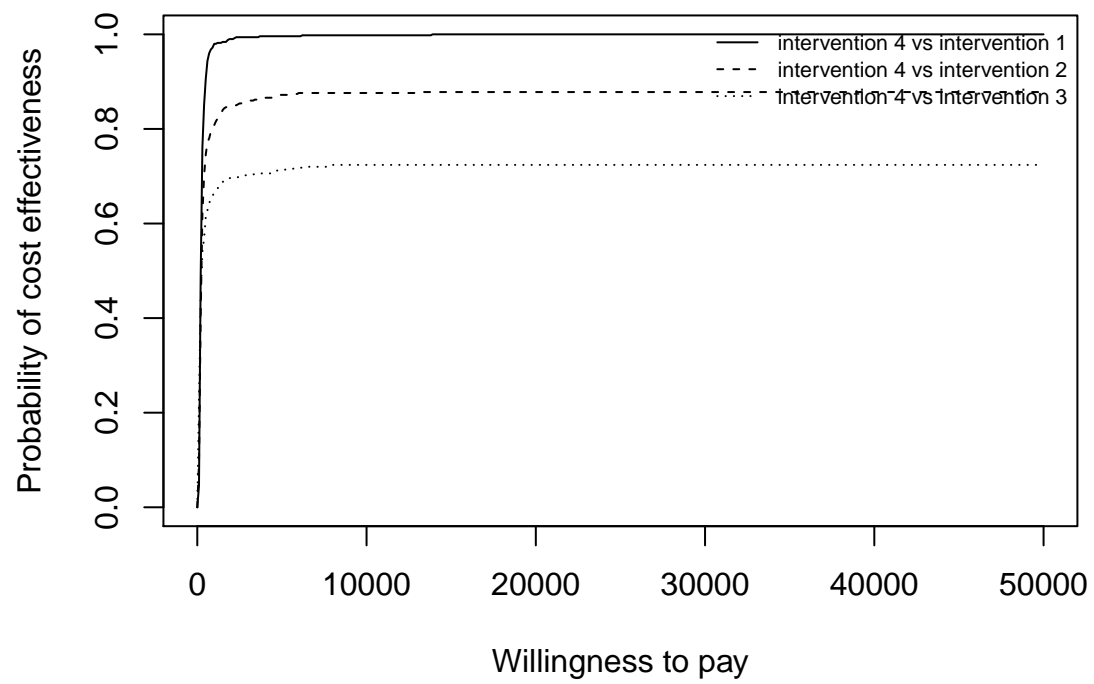
```
#> Wrong number of colours provided. Falling back to default
```

Cost Effectiveness Acceptability Curve

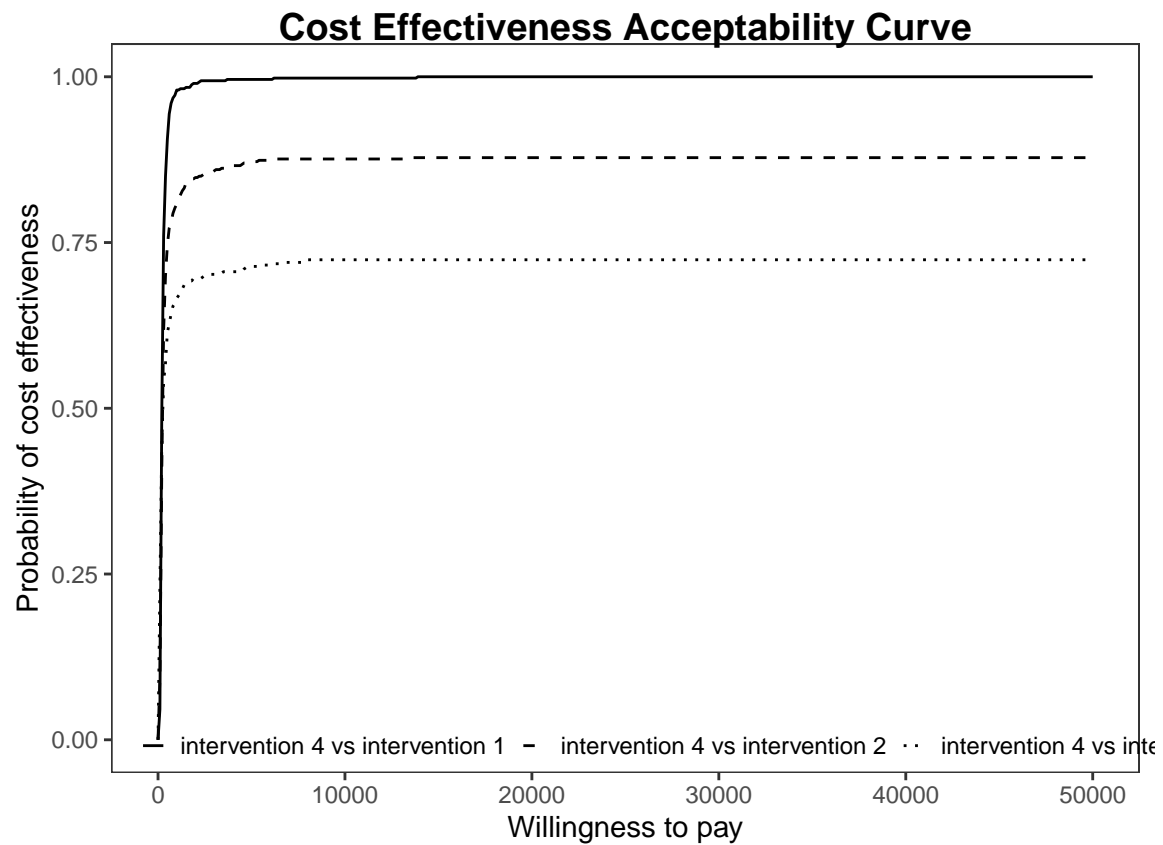


```
ceac.plot(he, pos = c(1, 1))  
#> Wrong number of colours provided. Falling back to default
```

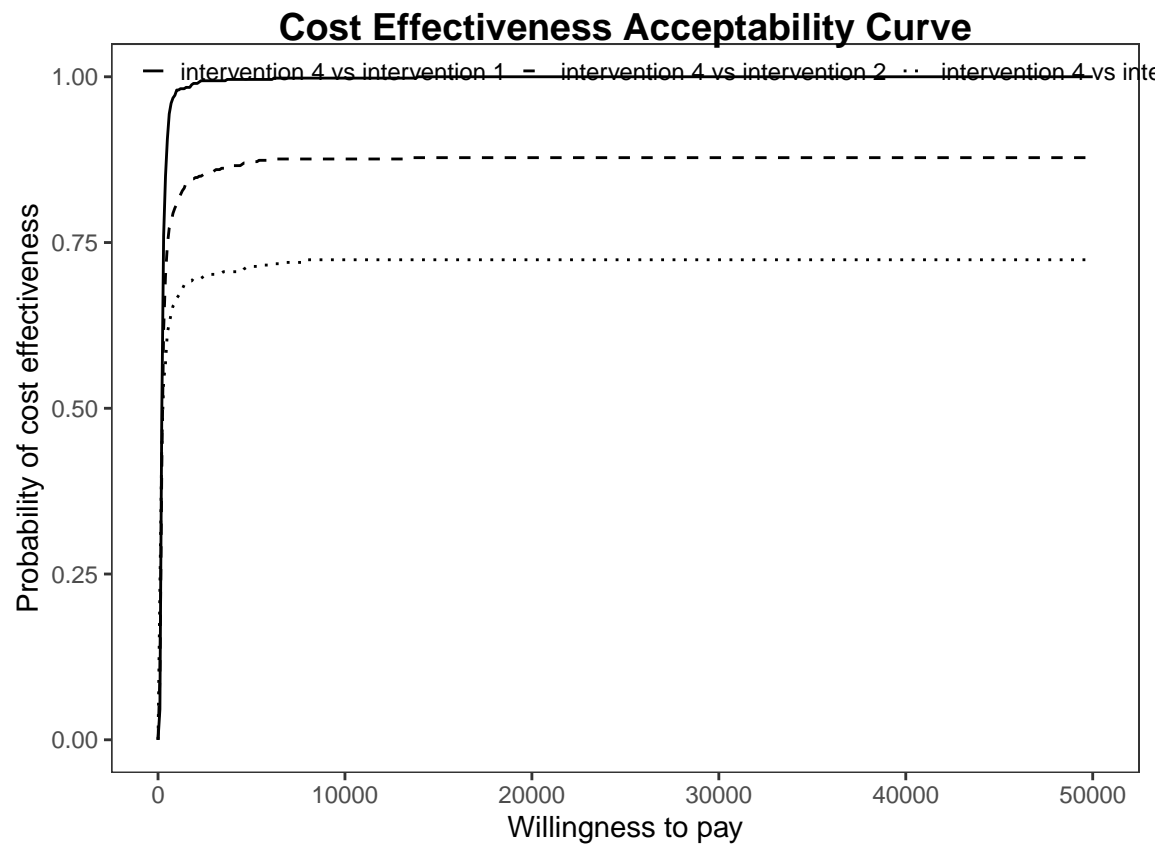
Cost Effectiveness Acceptability Curve



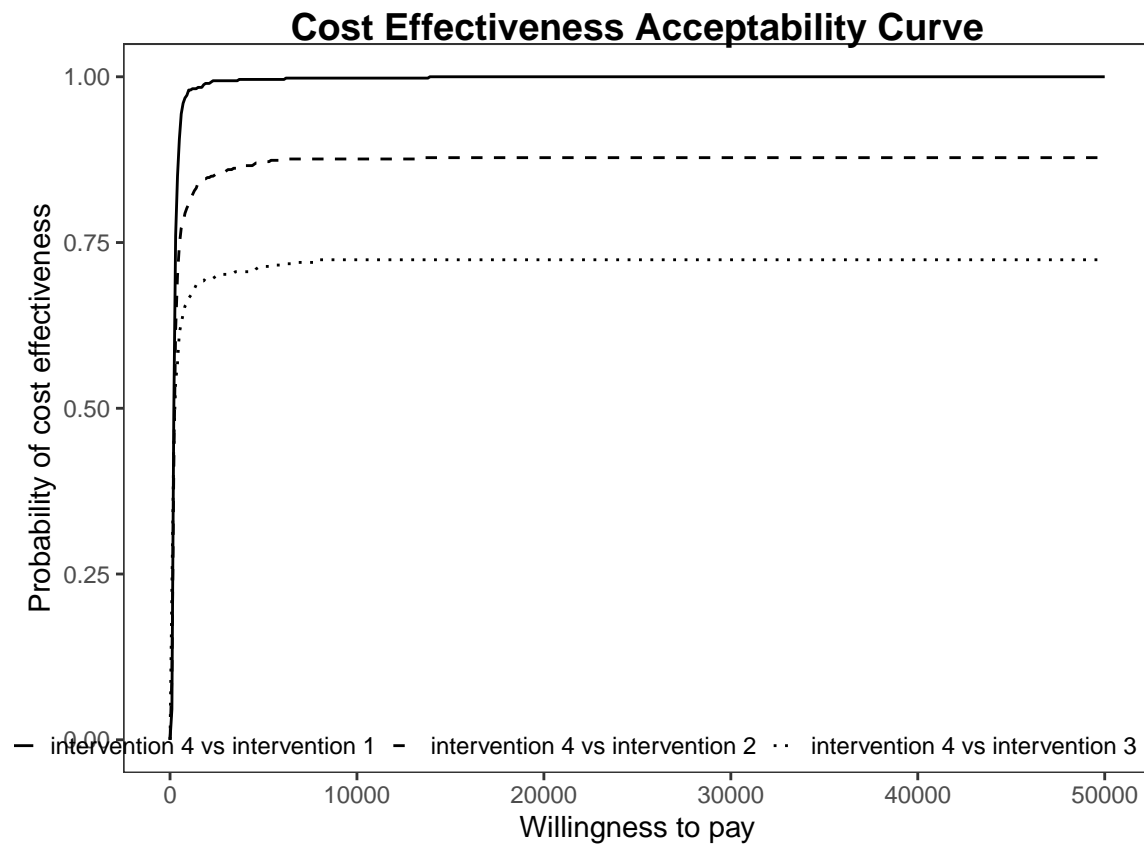
```
ceac.plot(he, graph = "ggplot2", pos = c(0, 0))
```



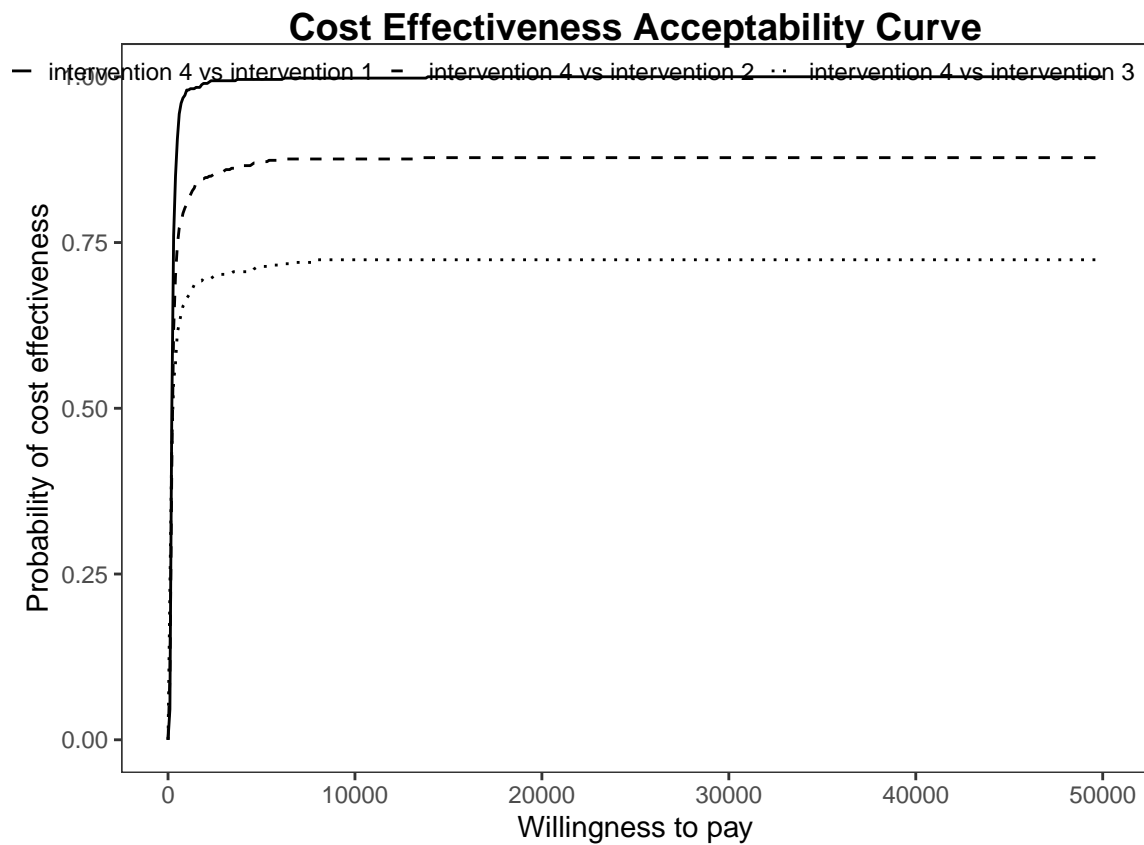
```
ceac.plot(he, graph = "ggplot2", pos = c(0, 1))
```



```
ceac.plot(he, graph = "ggplot2", pos = c(1, 0))
```



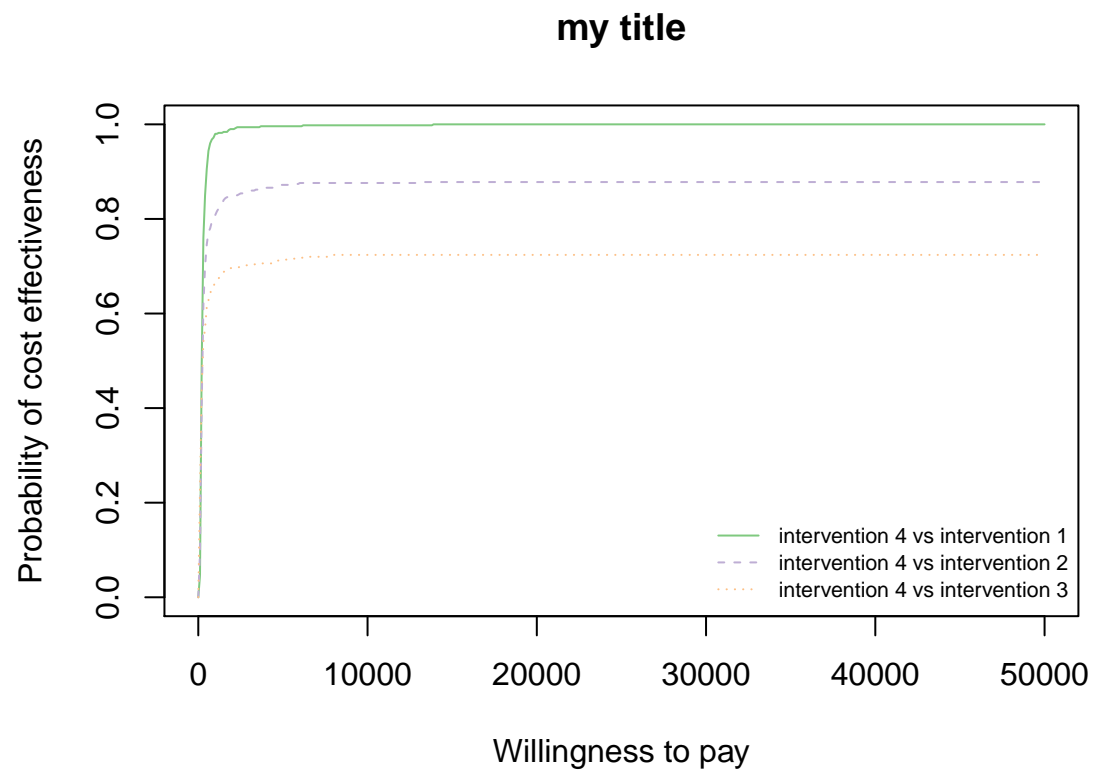
```
ceac.plot(he, graph = "ggplot2", pos = c(1, 1))
```



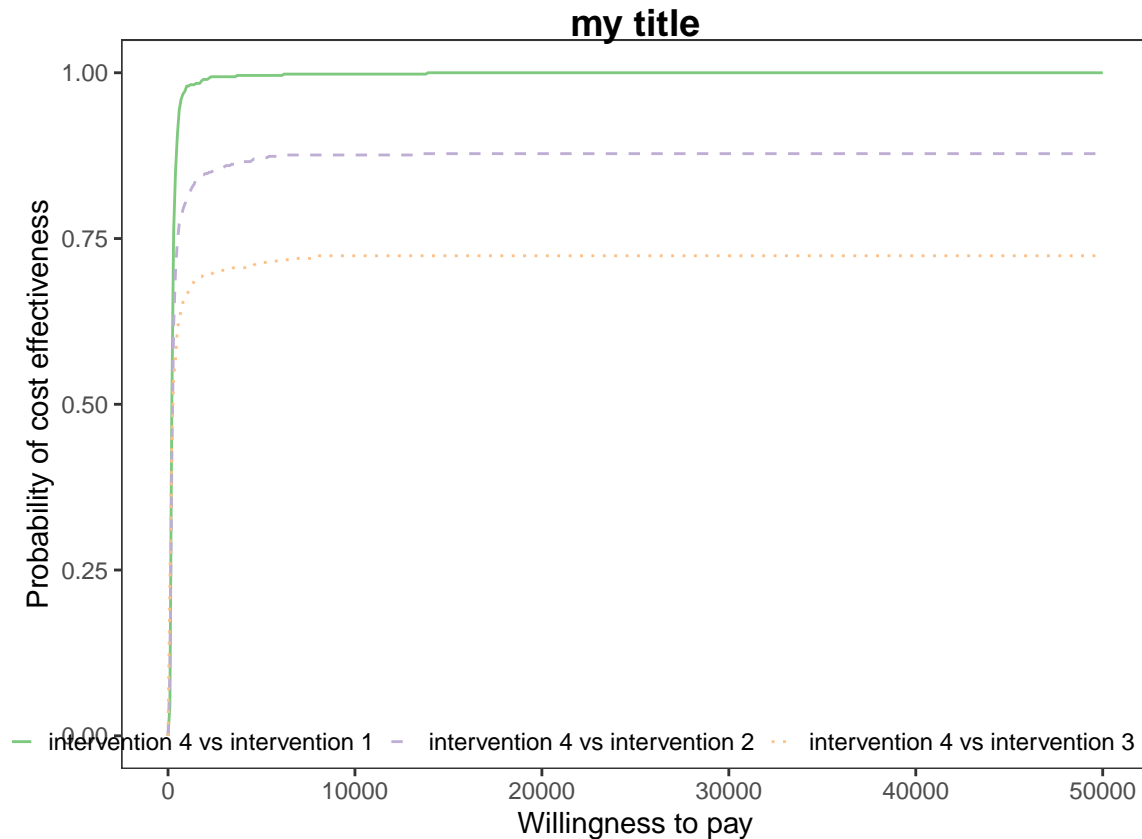
Define colour palette.

```
mypalette <- RColorBrewer::brewer.pal(3, "Accent")
```

```
ceac.plot(he,
  graph = "base",
  title = "my title",
  line = list(colors = mypalette),
  pos = FALSE)
```

```
ceac.plot(he,  
  graph = "ggplot2",  
  title = "my title",  
  line = list(colors = mypalette),  
  pos = FALSE)
```



Pair-wise comparisons

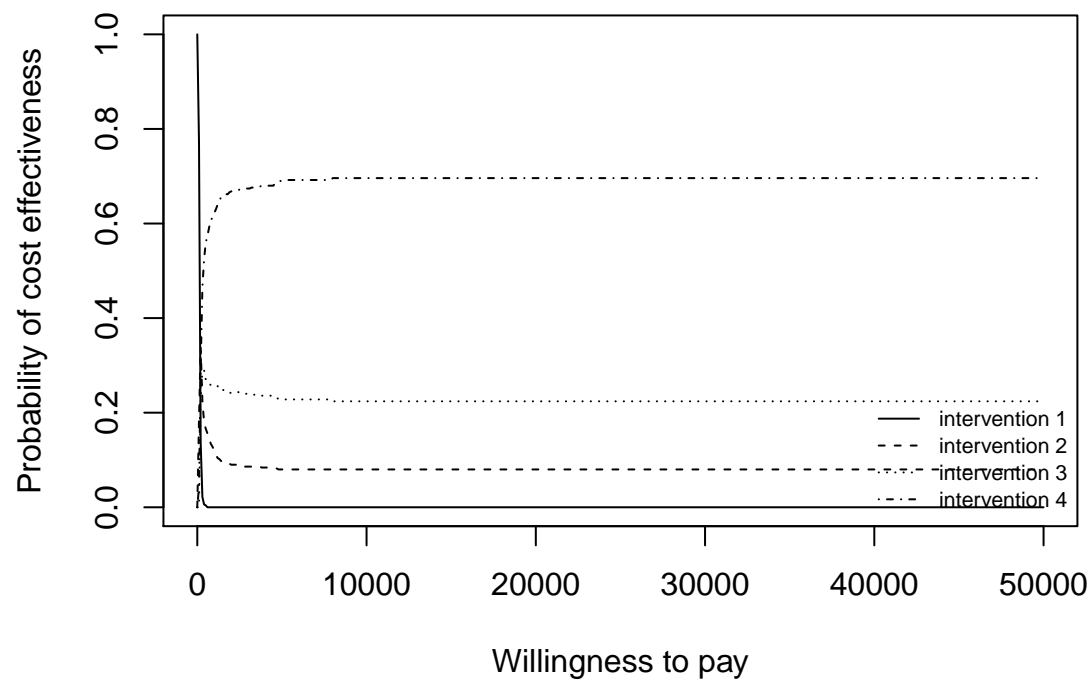
First we must determine all combinations of paired interventions using the `multi.ce()` function.

```
he <- multi.ce(he)
```

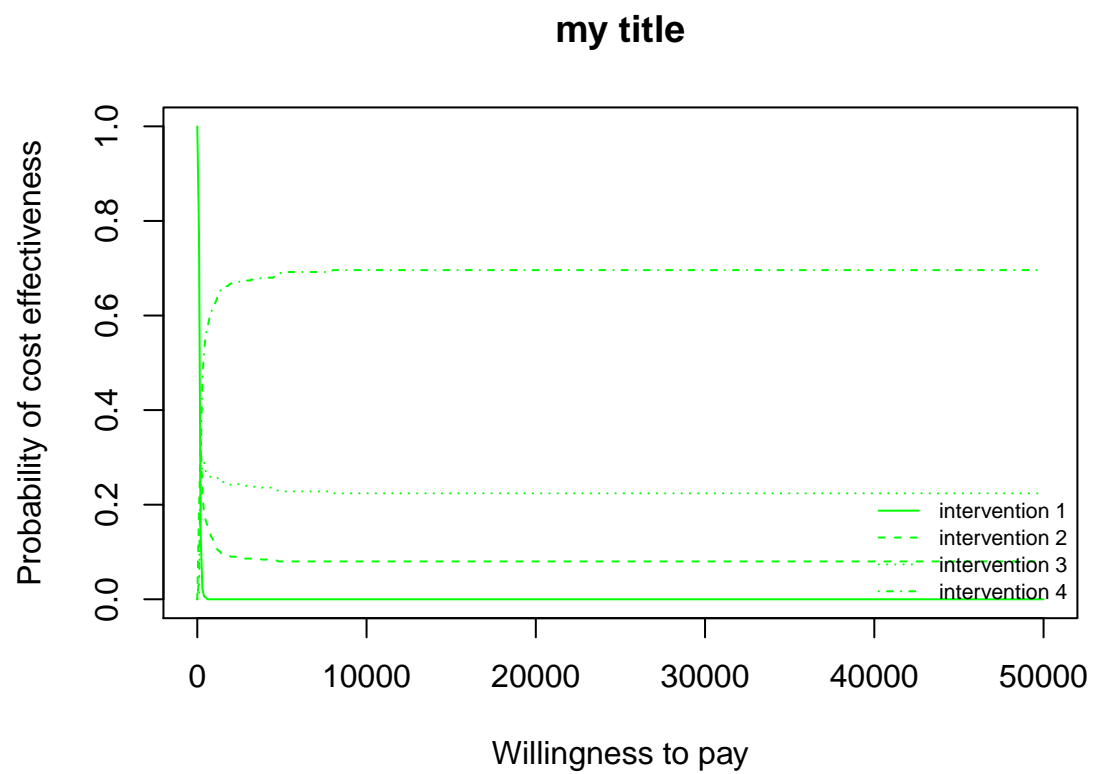
We can use the same plotting calls as before i.e. `ceac.plot()` and BCEA will deal with the pairwise situation appropriately. Note that in this case the probabilities at a given willingness to pay sum to 1.

```
ceac.plot(he, graph = "base")
#> Wrong number of colours provided. Falling back to default
```

Cost Effectiveness Acceptability Curve

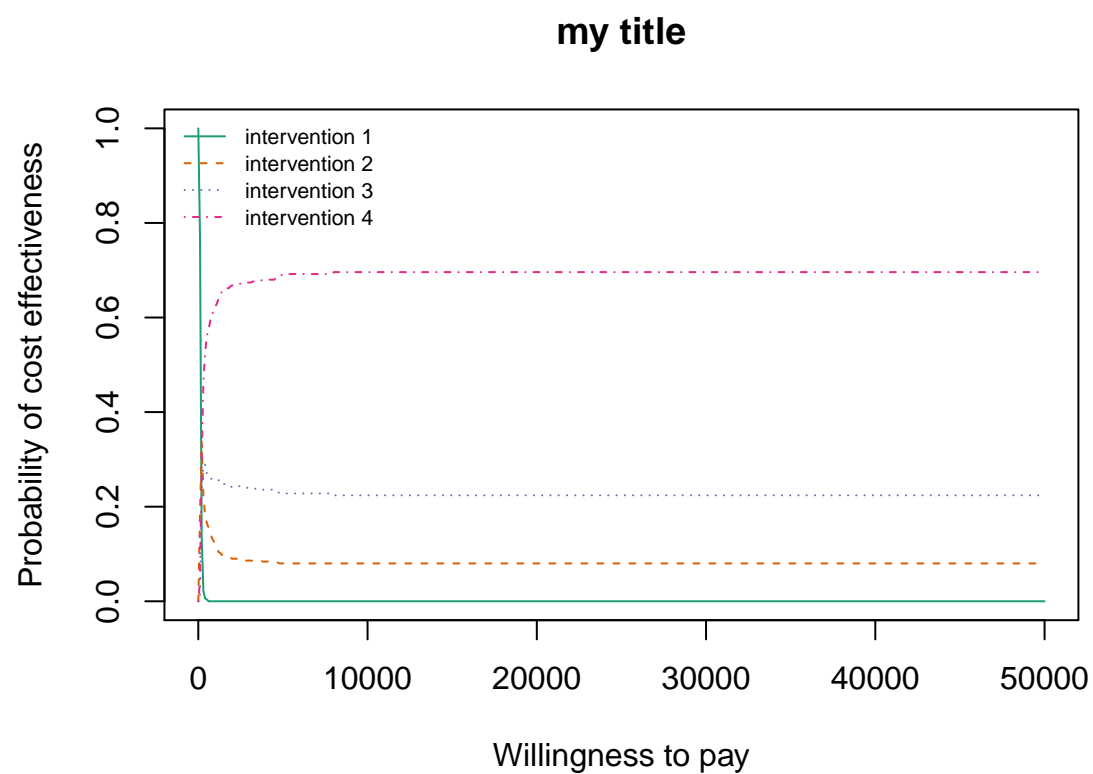


```
ceac.plot(he,  
  graph = "base",  
  title = "my title",  
  line = list(colors = "green"),  
  pos = FALSE)  
#> Wrong number of colours provided. Falling back to default
```



```
mypalette <- RColorBrewer::brewer.pal(4, "Dark2")

ceac.plot(he,
  graph = "base",
  title = "my title",
  line = list(colors = mypalette),
  pos = c(0,1))
```



```
ceac.plot(he,  
  graph = "ggplot2",  
  title = "my title",  
  line = list(colors = mypalette),  
  pos = c(0,1))
```

