# ThermogramBaseline Demo

Here we load the package and look at the data. Data must be formatted in this way:

```
library(ThermogramBaseline)
#> Loading required package: dplyr
#> Warning: package 'dplyr' was built under R version 4.1.3
#>
#> Attaching package: 'dplyr'
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
#> Loading required package: ggplot2
```

```
data(UrineWorking, package = "ThermogramBaseline")
head(UrineWorking)
```

| SampleID | Temperature | dCp |
|---|---:|---:|
| 1a | 20.00053 | -1.174067 |
| 1a | 20.01116 | -2.676789 |
| 1a | 20.01872 | -3.461386 |
| 1a | 20.02728 | -3.375907 |
| 1a | 20.03676 | -3.123136 |
| 1a | 20.04711 | -2.763205 |

Each function can only take in one thermogram sample at a time so we must filter for sample 1a and filter between the temperatures 45-90

```
SampleIDs <- UrineWorking %>% pull(SampleID) %>% unique() %>% as.vector()

Sample.1 <- UrineWorking %>% filter(SampleID == SampleIDs[1]) %>%
  filter(between(Temperature, 45,90))
```

Now that we have our first sample, lets get our two endpoints for baseline detection.

```
endpoints <- endpoint.detection(x = Sample.1, w = 90, exclusion.lwr = 60,
                                exclusion.upr = 80, point.selection = "innermost")
#> Scanning Lower.
#> Scanning Upper.
```

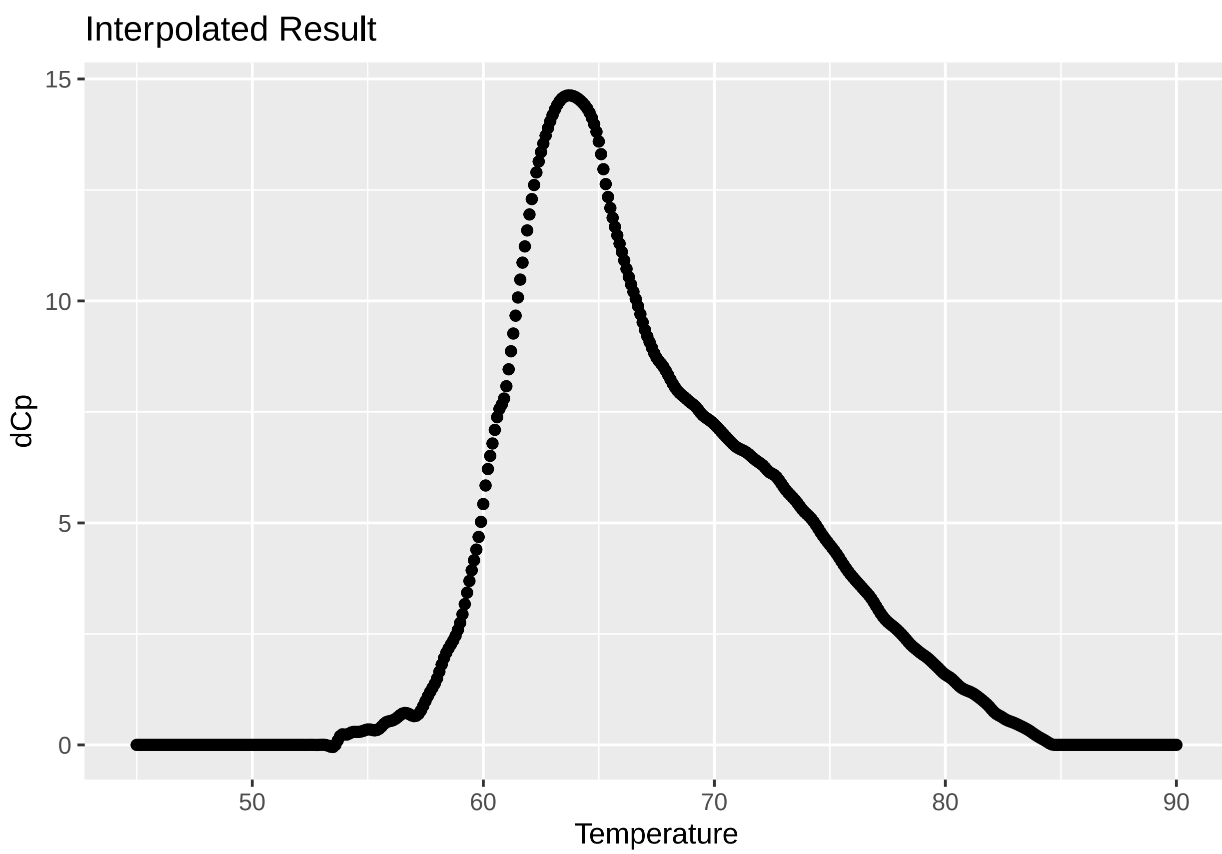
| lower | upper | method |
|---|---|---|
| 53.24153 | 84.63846 | innermost |

The model selected the temperatures 53.24153 and 84.63846 as our endpoints for our baseline subtraction. Next lets subtract the baseline using these endpoints.

```
baseline <- baseline.subtraction.byhand(x= Sample.1, lwr.temp = endpoints$lower,
                                        upr.temp = endpoints$upper, plot.on = FALSE)
```

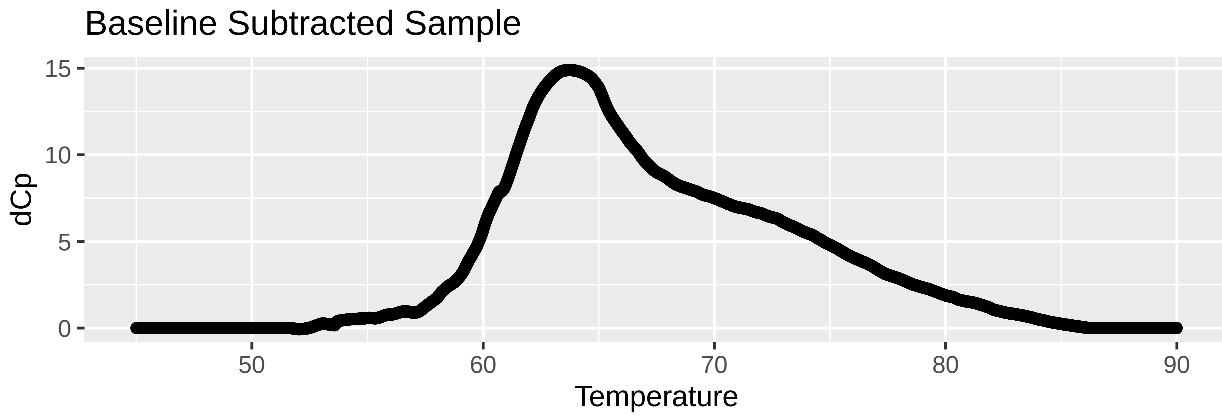
Now that we have subtracted the baseline, we will interpolate the data onto a grid of set temperatures. Notice plot.on is true by

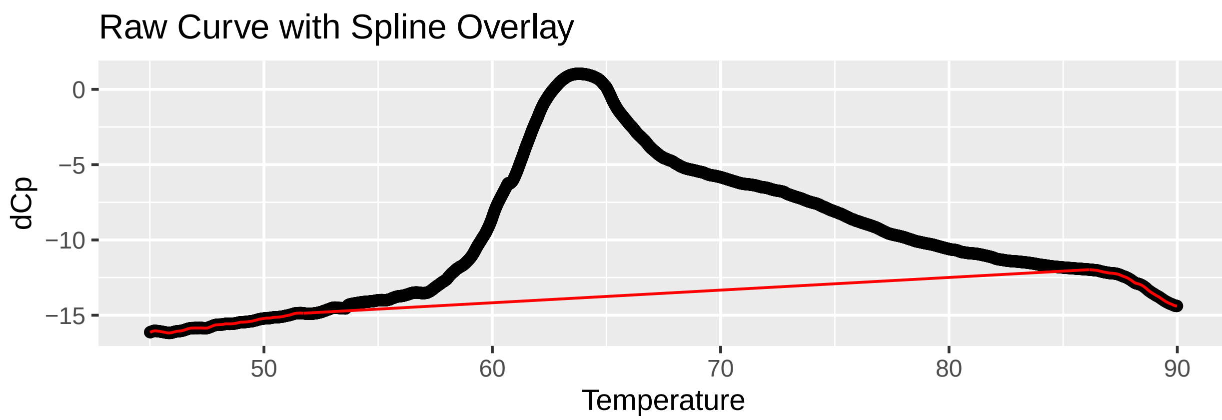
```
final <- final.sample.interpolate(x=baseline,grid.temp = seq(from = 45,to = 90,by = 0.1))
```

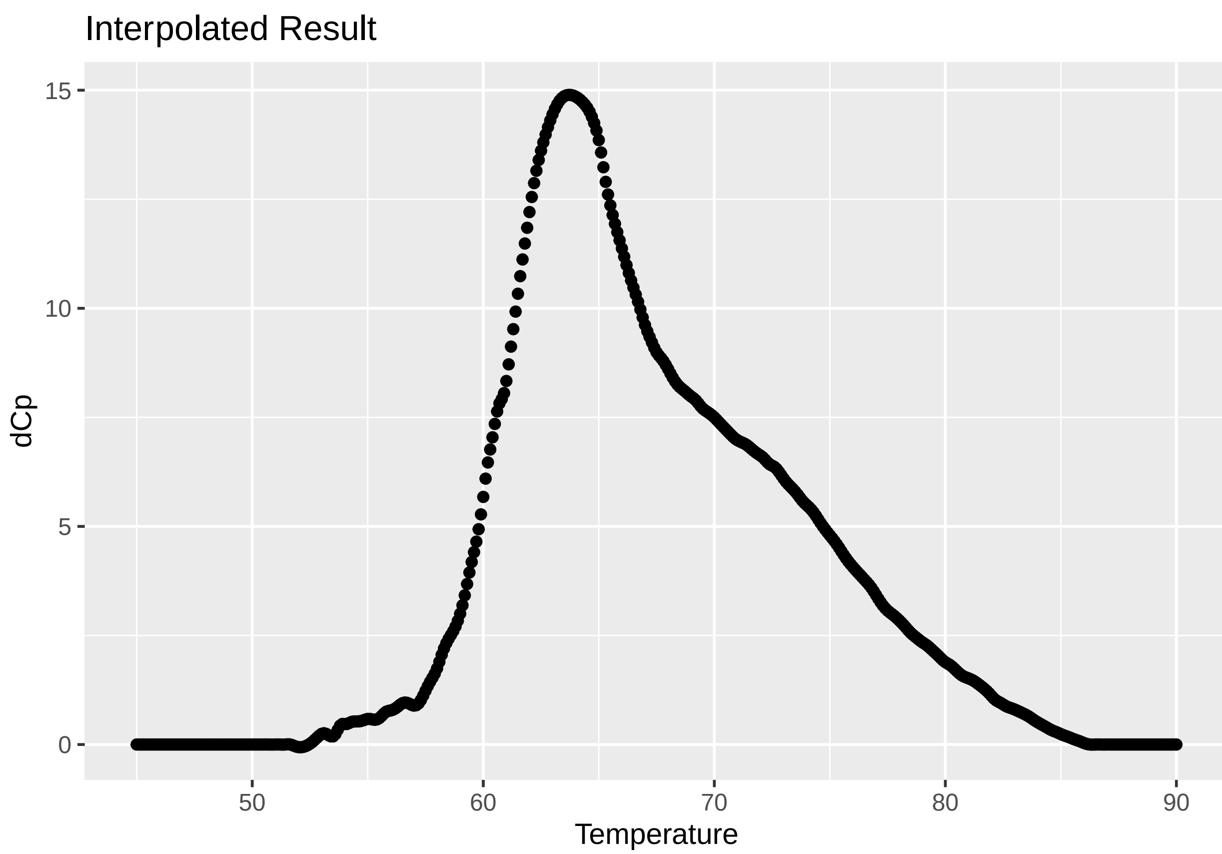



Here we have our final baseline subtracted interpolated sample. We can also use the auto function to complete these in one step. Notice that the plot.on default is false so we must set it to true if we want graphs to appear.

```
auto <- auto.baseline(x=Sample.1, w=90, exclusion.lwr = 60, exclusion.upr = 80,
                      grid.temp = seq(45,90,0.1),plot.on = TRUE)
#> Scanning Lower.
#> Scanning Upper.
```


Baseline Subtracted Sample
dCp
15
10
5
0
50
60
70
80
90
Temperature


Raw Curve with Spline Overlay
dCp
0
−5
−10
−15
50
60
70
80
90
Temperature




Lastly, lets run all 781 thermograms through our function and graph the end product. This code might take a while to process.

```
### Set our grid of temperatures to interpolate onto
grid.temp <- seq(from = 45,to = 90, by = 0.1)

### Create empty data frame to store results
Final.Results <- data.frame(Temperature = grid.temp)

### Store all sample IDs and how many samples we need to analyze.
All.IDs <- UrineWorking %>% pull(SampleID) %>% unique() %>% as.vector()
n.samples <- length(All.IDs)

### For loop for running all the samples through the algorithm
for(j in 1:n.samples)
{
  cat('Working on Sample ', All.IDs[j], 'element ', j,' of', n.samples,' \n')
  ### select a sample
  working.sample <- UrineWorking %>%
    filter(SampleID == All.IDs[j]) %>%
    select(Temperature, dCp)
  ### get a baseline-subtracted and interpolated final result!
  auto.output <- auto.baseline(x = working.sample, grid.temp = grid.temp)
  Final.Results <- Final.Results %>% cbind(out = auto.output$dCp)
  cat("\014")
}
```

```
### Rename the column names to the correct
colnames(Final.Results)[-1] <- All.IDs

### The below line will save the resulting data frame as a csv file
### Only run this line once
#write.csv(x = Final.Results, file = 'Final.Results.csv')
```

The code above provides a data frame with 781 baseline subtracted thermograms interpolated all to the same temperature grid. Below it is shown how to produce a graph of each thermogram output.

```
pdf('/Final.Results.pdf')
{
  for(j in 1:length(All.IDs))
  {
    g1 <- Final.Results %>% select(Temperature, All.IDs[j]) %>%
      ggplot(aes(x = Temperature, y = .[,2])) +
      geom_point() +
      labs(title = paste0('Final Automated Sample for ', str_sub(All.IDs[j], 2)))
    print(g1)
  }
}
dev.off()
```