pre_plots

Tingyu Zhu

12/1/2020

```r
library(tidyverse)
library(here)
library(ggplot2)
library(purrr)
devtools::load_all()
```

picture illustrating the kde method

```r
x <- c(1,2,8,10)
g <- seq(-5,20, 0.05)

kde_pic <- as.data.frame(f=map(x,~dnorm(.x-g)),g)

kde_pic <- as.data.frame(x) %>% mutate(
  f = map(x,~dnorm(.x-g)),
  grid = map(x,~g)
)

kde_pic <- unnest(kde_pic)
```

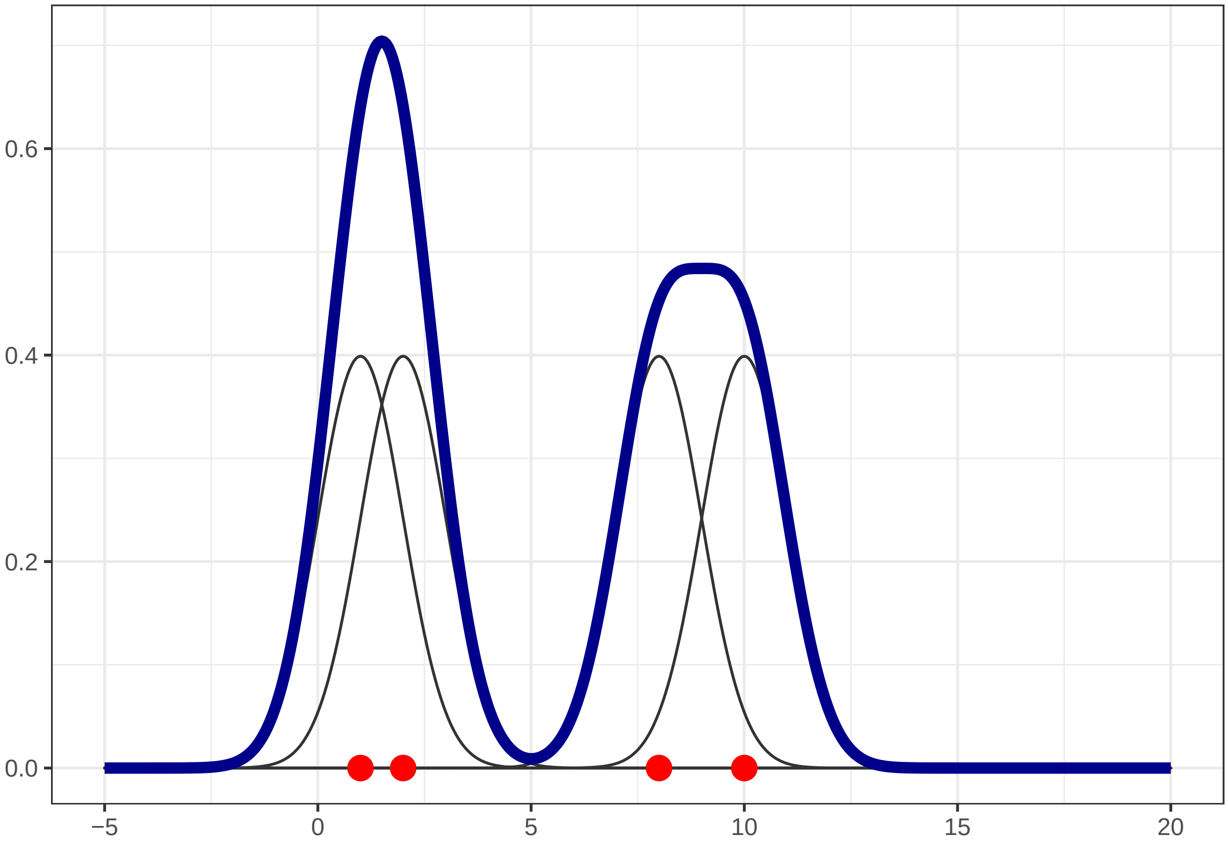
```
## Warning: `cols` is now required.
## Please use `cols = c(f, grid)`
```

```r
kde_pic_t <- data.frame(g=g, f=apply(matrix(kde_pic$f, nrow = length(g)), 1, sum))

pointdata <- data.frame( xname = x, ypos = c(0,0,0,0) )

#png(here("plots", "KDE_demo.png"),width=1200, height=600)

ggplot(kde_pic)+
 geom_path( aes(x = grid,y = f), color="gray20")+
 geom_line(data=kde_pic_t , aes(x = g,y = f), color="darkblue", size=2) +
geom_point(data = pointdata, mapping =
                    aes(x = xname, y = ypos), color="red", size=4)+
  xlab(NULL)+
  ylab(NULL)+
  theme_bw()
```



```
#dev.off()
```

Plots to illustrate the effect of the kernel function & bandwidth

```
# data setting
n <- 300
x <- rnorm(n)
grid <- seq(-4,4, 0.1)
```

Effect of the kernel function

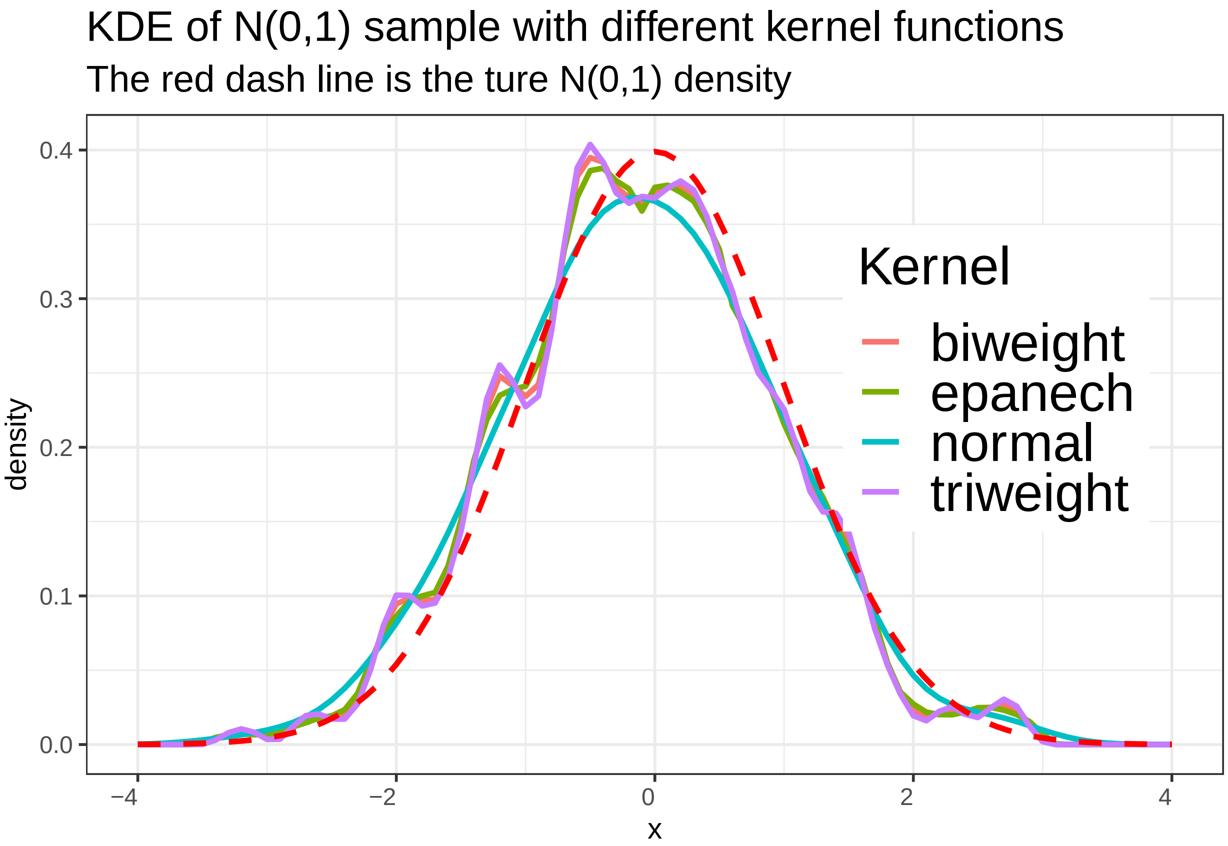
```
kers  <-  c("normal","epanech", "biweight", "triweight")
hs <- 1.06*sd(x)*n^(-0.2)

sim_ker <- as.data.frame(kers) %>% mutate(
  f_est = map(.x=kers,~KDE_est(x,grid,hs,.x)$f_est),
  grid = map(.x=kers,~KDE_est(x,grid,hs,.x)$grid)
)


## plots
sim_ker <- unnest(sim_ker)
```

```
## Warning: `cols` is now required.
## Please use `cols = c(f_est, grid)`
```

```
#png(here("plots", "KDE_kernels_smal_n.png"),width=1200, height=600)
sim_ker %>%
  ggplot()+
  geom_line(aes(x = grid,y = f_est,color = kers), size=1)+
  stat_function(fun = dnorm,
                args = list(mean =0, sd = 1),
                color = "red", linetype = "dashed", size=1) +
  xlab("x")+
  ylab("density")+
  theme_bw() +
  labs(title = "KDE of N(0,1) sample with different kernel functions",
       subtitle = "The red dash line is the ture N(0,1) density",
       color="Kernel")+
  theme(axis.text.x = element_text( hjust = 1),
        plot.title = element_text(size=16),
        plot.subtitle = element_text(size=14))+
   theme(legend.position = c(0.8, 0.6),
         legend.title = element_text(size = 20),
          legend.text = element_text(size = 20))
```



```
#dev.off()
```

Effect of the bandwidth

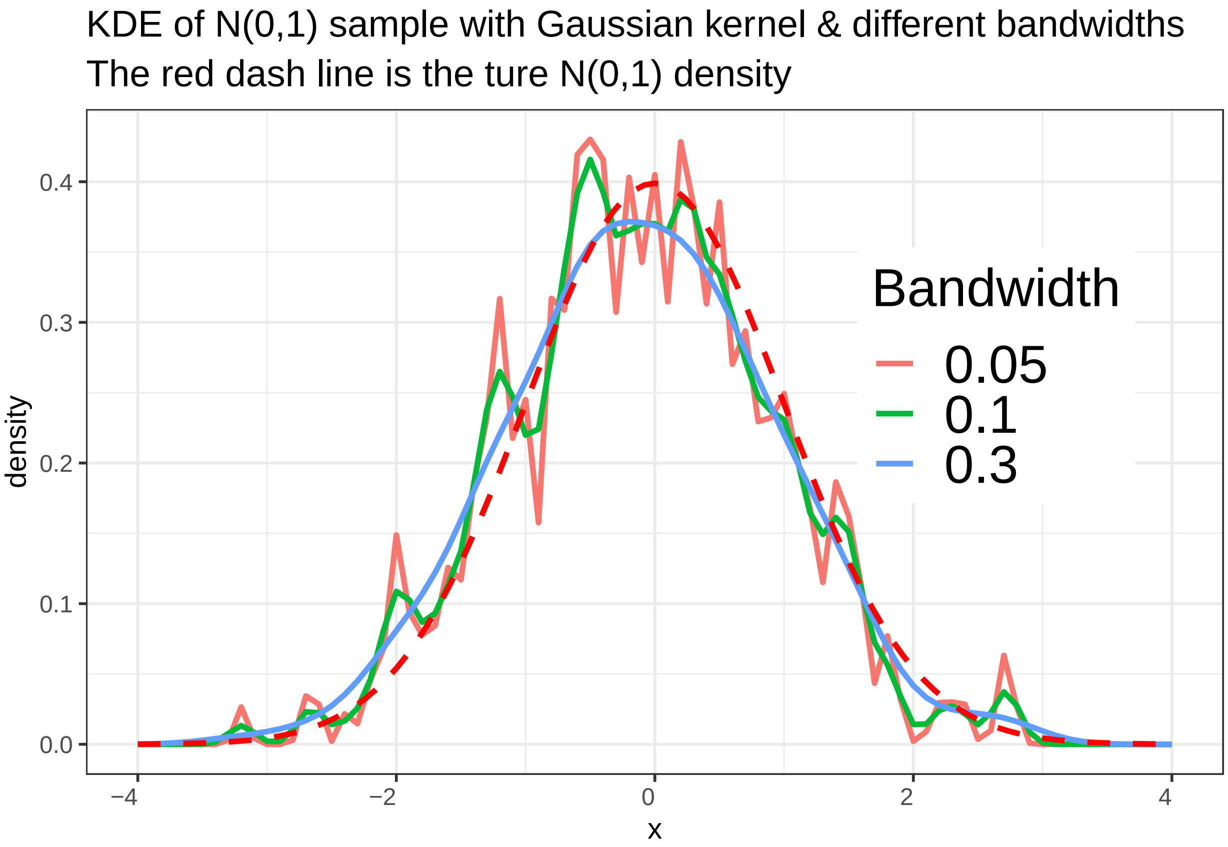
```r
h <- c(0.05,0.1,0.3)

sim_h <- as.data.frame(h) %>% mutate(
  f_est = map(.x=h,~KDE_est(x,grid,.x,"normal")$f_est),
  grid = map(.x=h,~KDE_est(x,grid,.x,"normal")$grid)
)

## plots
sim_h <- unnest(sim_h)
```

```
## Warning: `cols` is now required.
## Please use `cols = c(f_est, grid)`
```

```r
#png(here("plots", "KDE_bandwidths_small_n.png"),width=1200, height=600)
sim_h %>%
  ggplot()+
  geom_line(aes(x = grid,y = f_est,color = factor(h)), size=1)+
  stat_function(fun = dnorm,
                args = list(mean =0, sd = 1),
                color = "red", linetype = "dashed", size=1) +
  xlab("x")+
  ylab("density")+
  theme_bw() +
  labs(title = "KDE of N(0,1) sample with Gaussian kernel & different bandwidths",
       subtitle = "The red dash line is the ture N(0,1) density",
       color = "Bandwidth")+
  theme(axis.text.x = element_text( hjust = 1),
        plot.title = element_text(size=14),
        plot.subtitle = element_text(size=14))+
   theme(legend.position = c(0.8, 0.6),
         legend.title = element_text(size = 20),
          legend.text = element_text(size = 20))
```

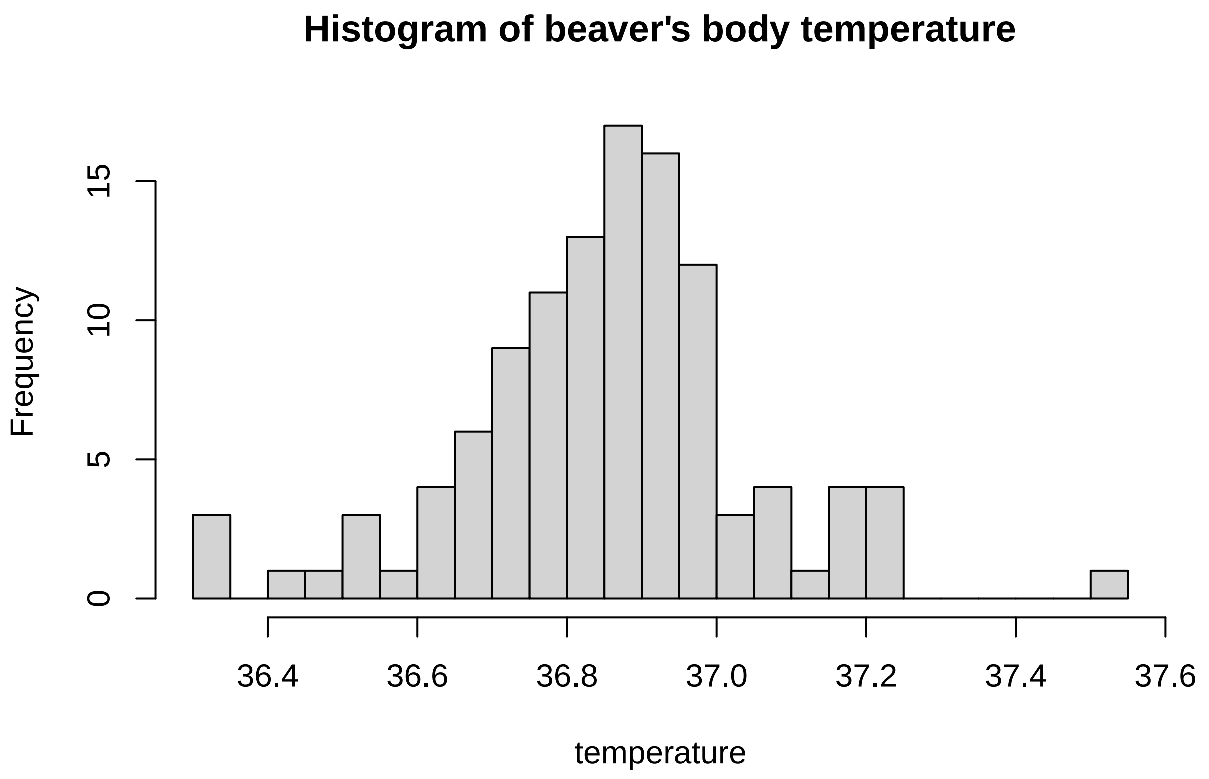

KDE of N(0,1) sample with Gaussian kernel & different bandwidths
The red dash line is the ture N(0,1) density

```
#dev.off()
```

Beaver’s data

Density plots using the bulit-in density function

```
#png(here("plots", "beaver_hist.png"),width=1200, height=600)
hist(beaver1$temp, breaks = 30, main = "Histogram of beaver's body temperature",
     xlab = "temperature")
```



```r
dev.off()
```

```
## null device
##           1
```

```r
png(here("plots", "beaver_density.png"),width=1200, height=600)
plot(density(beaver1$temp),
     main = "Estimation results using the built in density() function",
     xlab = "temperature")
#dev.off()
```

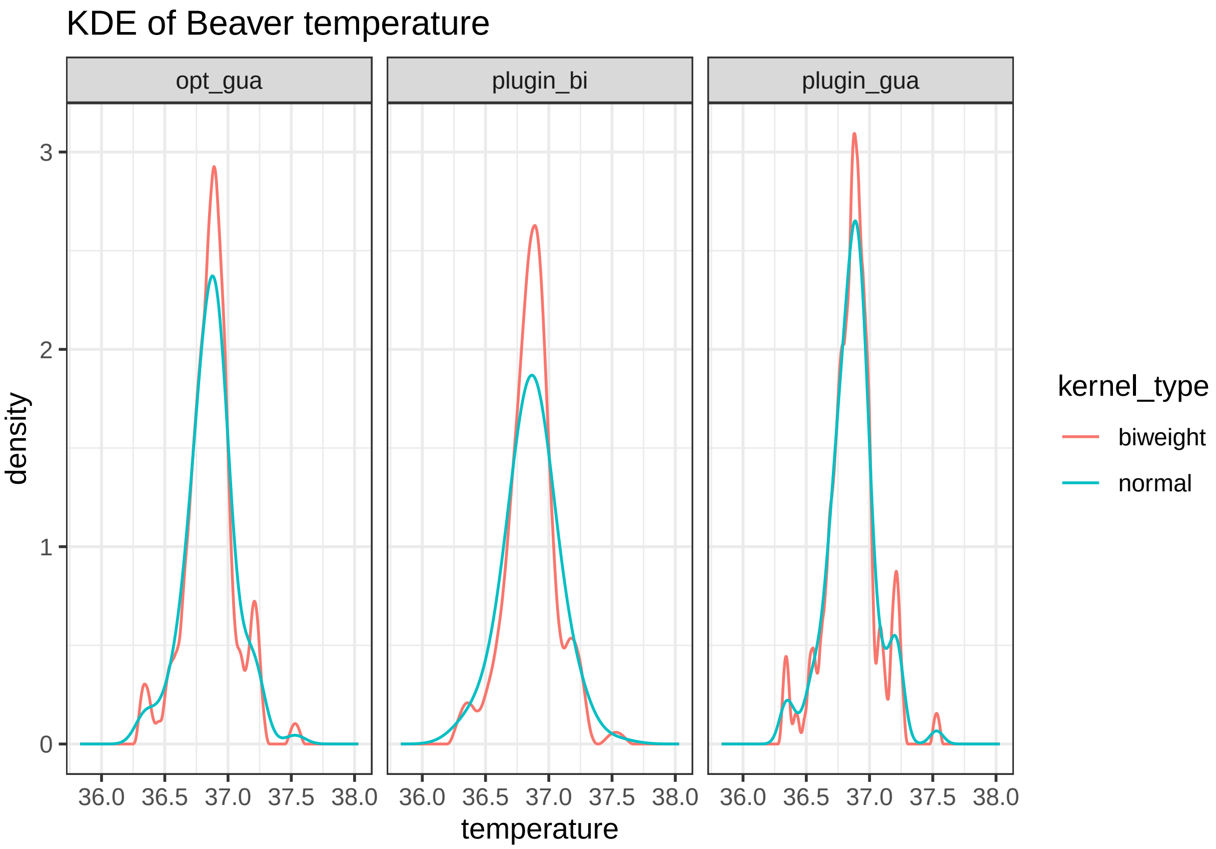
KDE estimation results using own function

```r
est_big <- read_rds(here("results", "beaver.rds"))

est_big_df <- unnest(est_big)
```

```
## Warning: `cols` is now required.
## Please use `cols = c(f_ests, grid)`
```

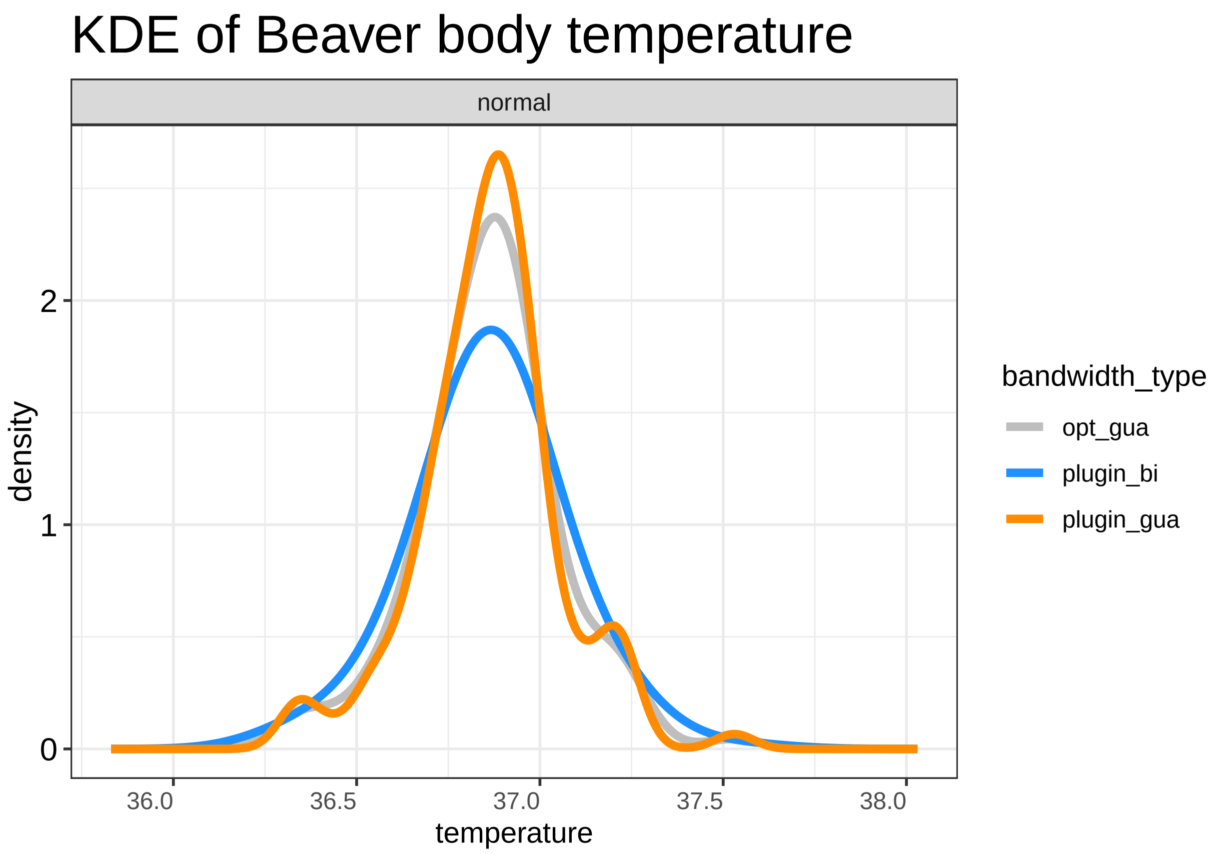
```r
est_big_df %>%
  ggplot()+
  geom_line(aes(x = grid,y = f_ests,color = kernel_type))+
  facet_wrap(~bandwidth_type)+
  xlab("temperature")+
  ylab("density")+
  theme_bw() +ggtitle("KDE of Beaver temperature")
```



```
method.c <- c("plugin_gua" = "darkorange",
              "plugin_bi" = "dodgerblue",
              "opt_gua" = "gray")

#png(here("plots", "beaver_KDE.png"),width=1200, height=600)

est_big_df %>% filter(kernel_type=="normal") %>%
  ggplot()+
  geom_line(aes(x = grid,y = f_ests,color = bandwidth_type), size=1.5)+
  facet_wrap(~kernel_type)+
  scale_colour_manual(values = method.c) +
   xlab("temperature")+
  ylab("density")+
  theme_bw() +ggtitle("KDE of Beaver body temperature")+
  theme(axis.text.x = element_text( hjust = 1),
        plot.title = element_text(size=20))+
  theme(axis.text.y = element_text(colour = 'black', size = 12),
          axis.title.y = element_text(size = 12,
          hjust = 0.5, vjust = 0.2))
```


KDE of Beaver body temperature
normal
density
2
1
0
36.0
36.5
37.0
37.5
38.0
temperature
bandwidth_type
opt_gua
plugin_bi
plugin_gua

```r
#dev.off()
```