Summary_week7

Tingyu Zhu

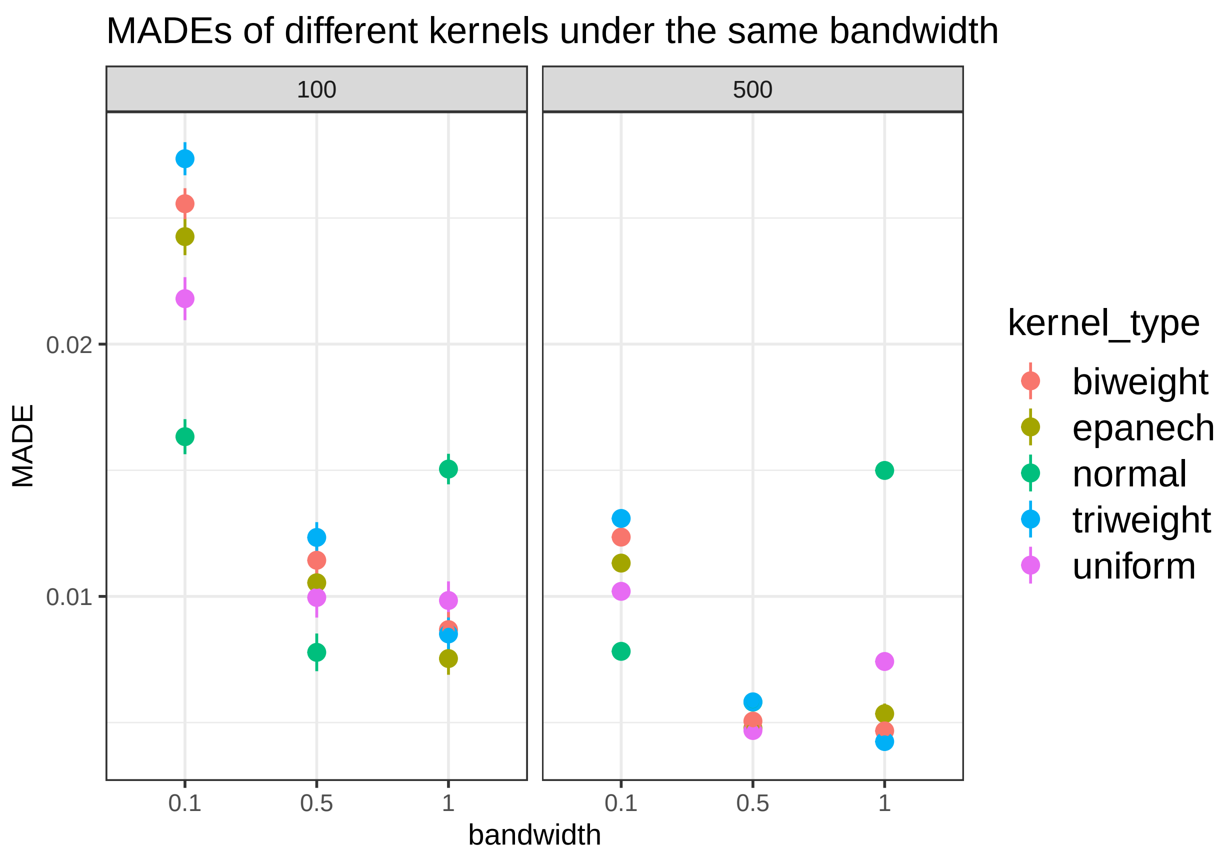
12/6/2020

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

Simulation result with different sample sizes & kernels & bandwidths(randomly selected)

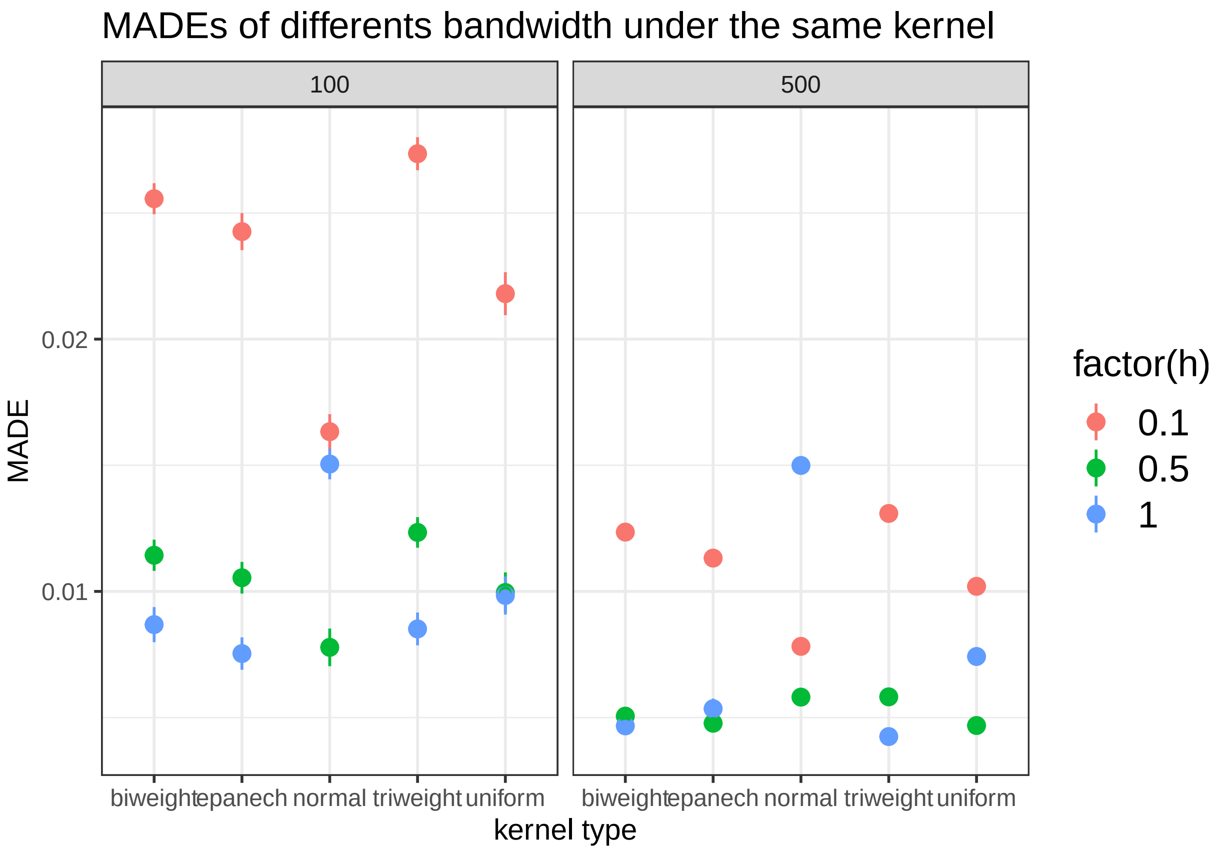
```r
made_comapre <- read_rds(here("results", "week7-sim-1-plot.rds"))
#png(here("plots", "KDE_big_xkernel.png"),width=1600, height=900)

made_comapre %>%
  ggplot(aes(x = factor(h), y = made_mean, color = kernel_type)) +
    geom_pointrange(aes(
      ymin = made_mean - 1.96*made_sd/sqrt(n.sim),
      ymax = made_mean + 1.96*made_sd/sqrt(n.sim))) +
    geom_point()  +
  facet_wrap(~ n) +
   theme_bw() +ggtitle("MADEs of different kernels under the same bandwidth")+
  xlab("bandwidth")+
  ylab("MADE")+
  theme( legend.title = element_text(size = 14),
          legend.text = element_text(size = 14),
         plot.title = element_text(size=14))
```




```
#dev.off()

#png(here("plots", "KDE_big_xbandwidth.png"),width=1600, height=900)
made_comapre %>%
  ggplot(aes(x =kernel_type, y = made_mean, color =  factor(h))) +
    geom_pointrange(aes(
      ymin = made_mean - 1.96*made_sd/sqrt(n.sim),
      ymax = made_mean + 1.96*made_sd/sqrt(n.sim))) +
    geom_point()  +
  facet_wrap(~ n)+
   theme_bw() +ggtitle("MADEs of differents bandwidth under the same kernel")+
    xlab("kernel type")+
  ylab("MADE")+
  theme( legend.title = element_text(size = 14),
          legend.text = element_text(size = 14),
         plot.title = element_text(size=14))
```




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

From the abvoe two plots, we can get the following conclusion:

1. Large sample size tends to have smaller MADE, that is, smaller estiation biases.
2. Under the same bandwidth, some of MADEs of different kernel functions are close to each other. That is, their estimation results using will be simialr.
3. Under the same kernel function, the MADEs with different chioce of bandwith are quite different.
4. There's no guarrantee that a perticular kernel or bandwith is better than others. To achieve the best estmation results, we need to find the best combinations of kernel and bandwidth. And large sample size will achieve more accurate estiamtion.

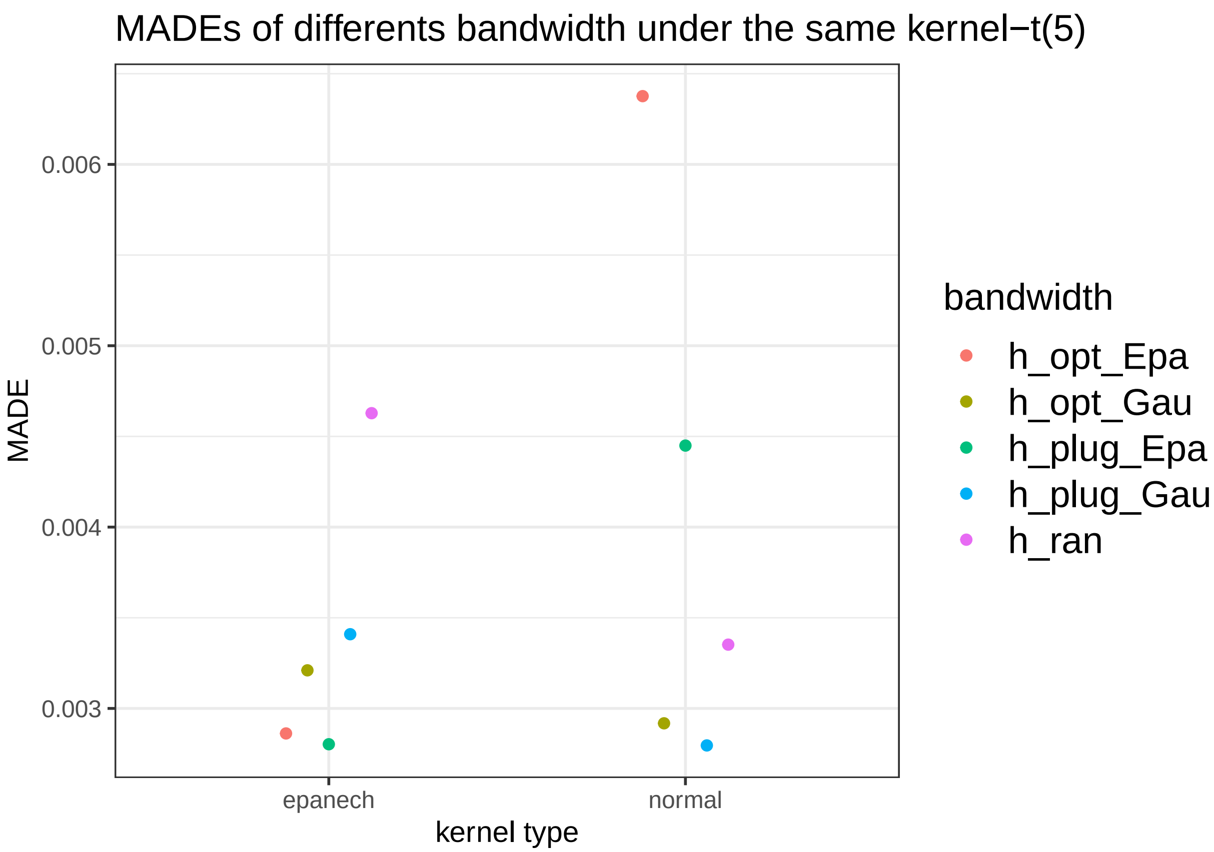
Simulation results for different distribution with differnts ways of choosing the optimal bandwidth

t(5)

```
comapre_bdwt_t <- read_rds(here("results", "week78-sim-t-plot.rds"))

#png(here("plots", "KDE_t.png"),width=1600, height=900)
comapre_bdwt_t %>%
ggplot(aes(x =kernel_type, y = made_mean, color =  bandwidth)) +
  geom_point(position=position_dodge(width=0.3))    +
```

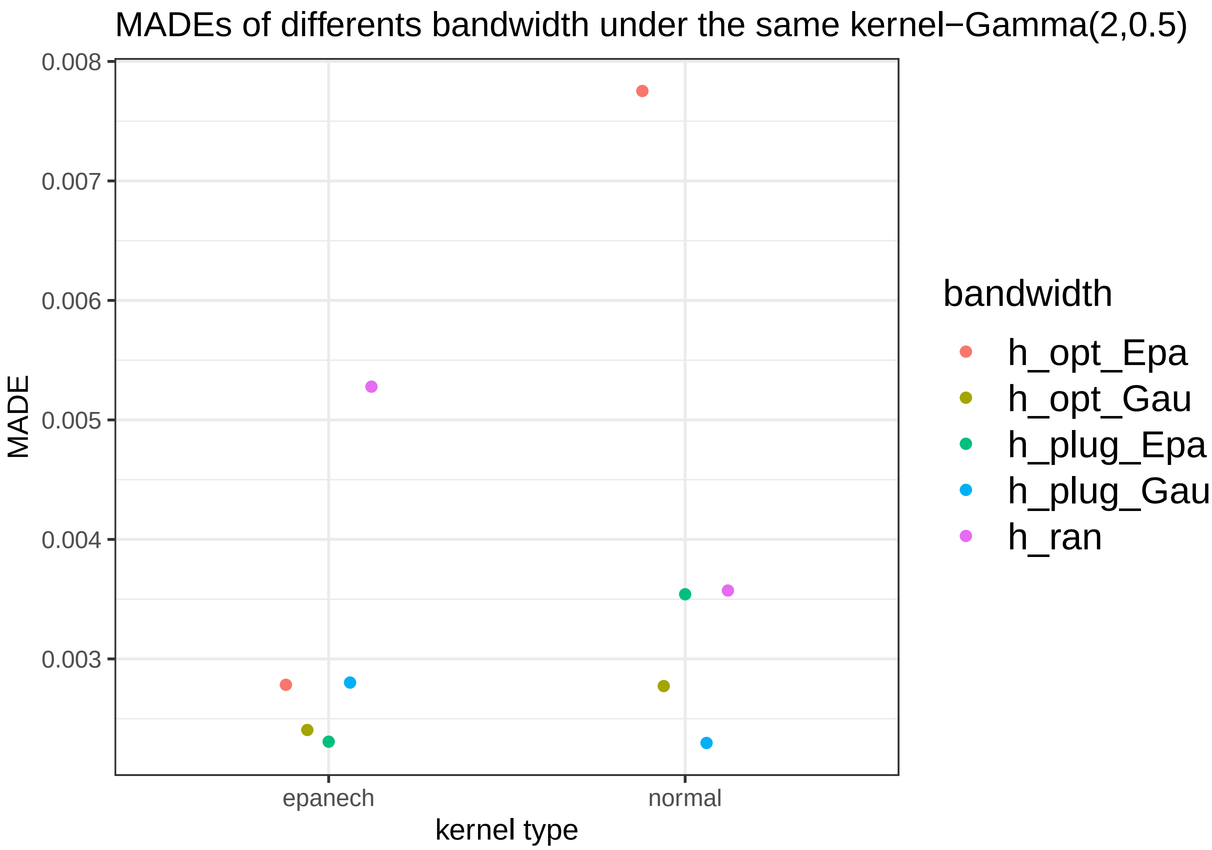
```r
  theme_bw() + ggtitle("MADEs of differents bandwidth under the same kernel-t(5)")+
   xlab("kernel type")+ylab("MADE")+
 theme( legend.title = element_text(size = 14),
         legend.text = element_text(size = 14),
        plot.title = element_text(size=14))
```




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

Gamma(2,0.5)

```r
comapre_bdwt_gamma <- read_rds(here("results", "week78-sim-gamma-plot.rds"))
#png(here("plots", "KDE_gamma.png"),width=1600, height=900)
comapre_bdwt_gamma %>%
ggplot(aes(x =kernel_type, y = made_mean, color = bandwidth)) +
  geom_point(position=position_dodge(width=0.3))   +
   theme_bw() + ggtitle("MADEs of differents bandwidth under the same kernel-Gamma(2,0.5)")+
    xlab("kernel type")+ylab("MADE")+
  theme( legend.title = element_text(size = 14),
          legend.text = element_text(size = 14),
         plot.title = element_text(size=13))
```



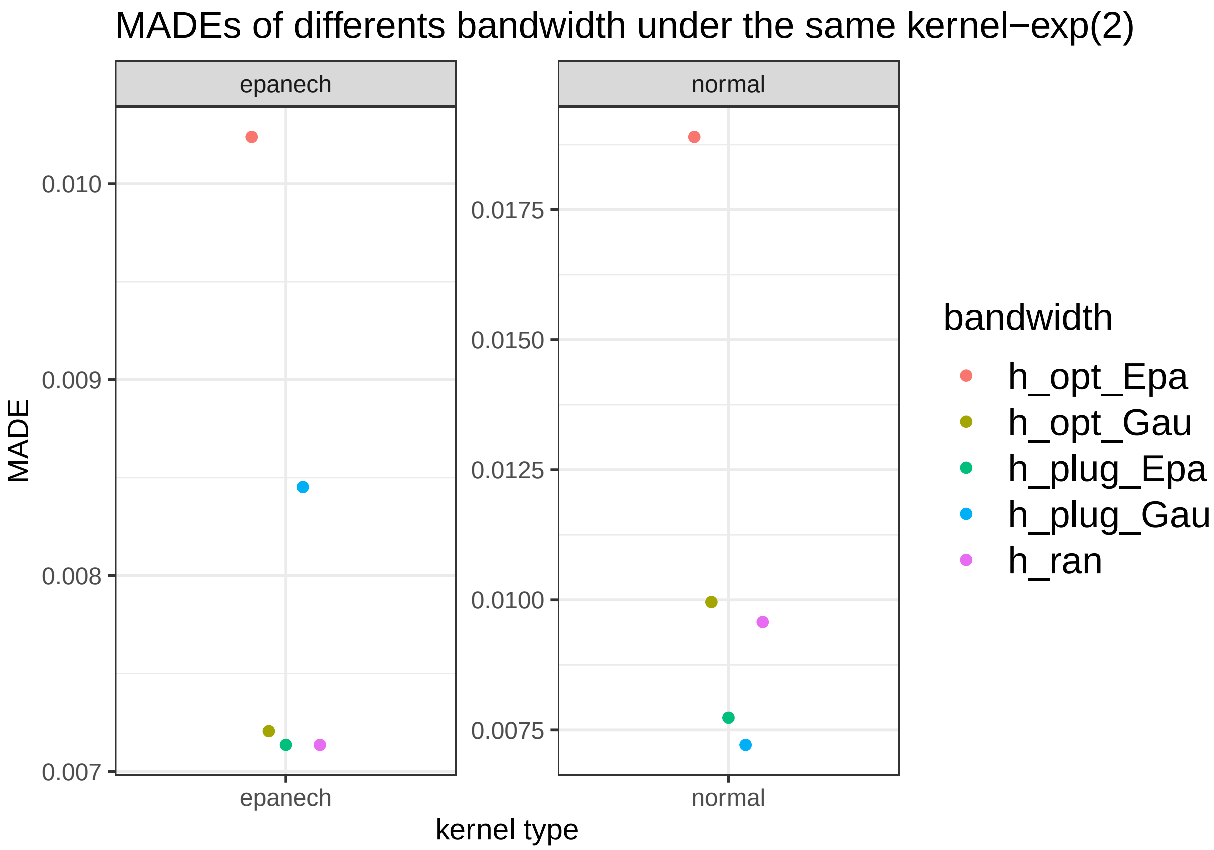

```
#dev.off()
```

exp(2)

```
comapre_bdwt_exp <- read_rds(here("results", "week78-sim-exp-plot.rds"))

#png(here("plots", "KDE_exp.png"),width=1600, height=900)
comapre_bdwt_exp %>%
ggplot(aes(x =kernel_type, y = made_mean, color =  bandwidth)) +
      geom_point(position=position_dodge(width=0.3)) +

    facet_wrap(~ kernel_type, scales ="free")+
   theme_bw() + ggtitle("MADEs of differents bandwidth under the same kernel-exp(2)")+
    xlab("kernel type")+ylab("MADE")+
  theme( legend.title = element_text(size = 14),
          legend.text = element_text(size = 14),
        plot.title = element_text(size=14))
```




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

Results show that under different distributions, the plug-in bandwidths may or may not have the smallest MADE (under the corresponding kernels). If the data is quite close to normal disreibution, may be the Normal reference bandwidth selector is the best chioce. If the data is far away from the normal distribution, the plug-in bandwidth selector is a better chioce. So, how to choose the best bandwidth with the most accurate estimation result is still a tough task.