In [1]:

```
library('kedd')
library('np')
```

```
Nonparametric Kernel Methods for Mixed Datatypes (version 0.60-10)
[vignette("np_faq",package="np") provides answers to frequently asked
questions]
[vignette("np",package="np") an overview]
[vignette("entropy_np",package="np") an overview of entropy-based meth
ods]
```

In [2]:

```
options(repr.plot.width = 10, repr.plot.height = 6)
```

In [3]:

```
?density
```

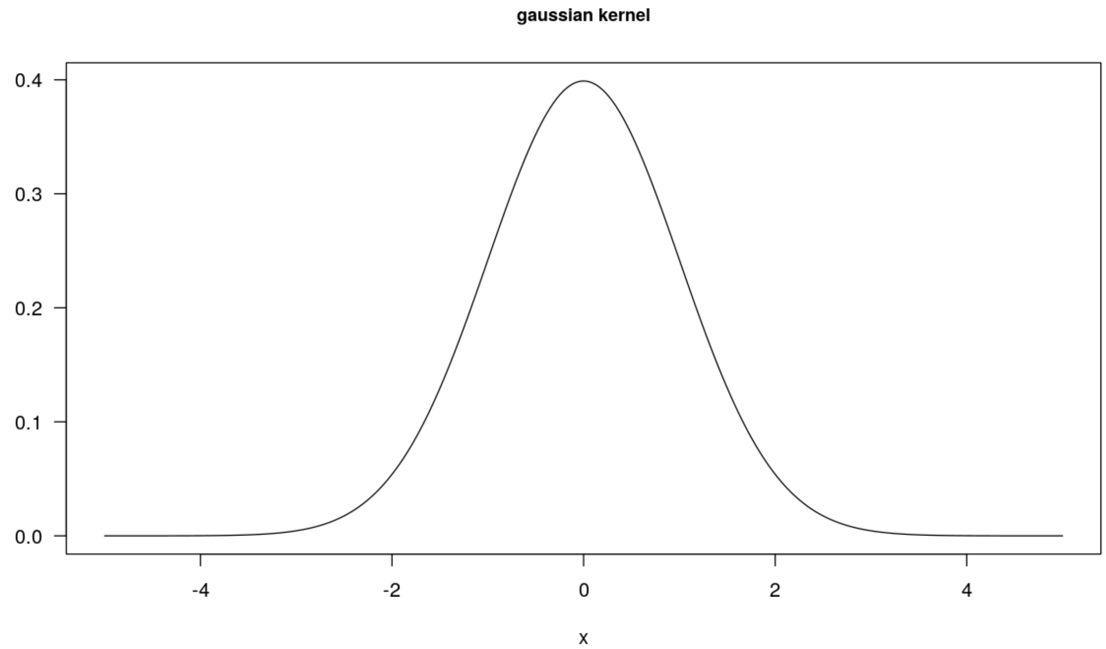
Ядерные оценки плотности

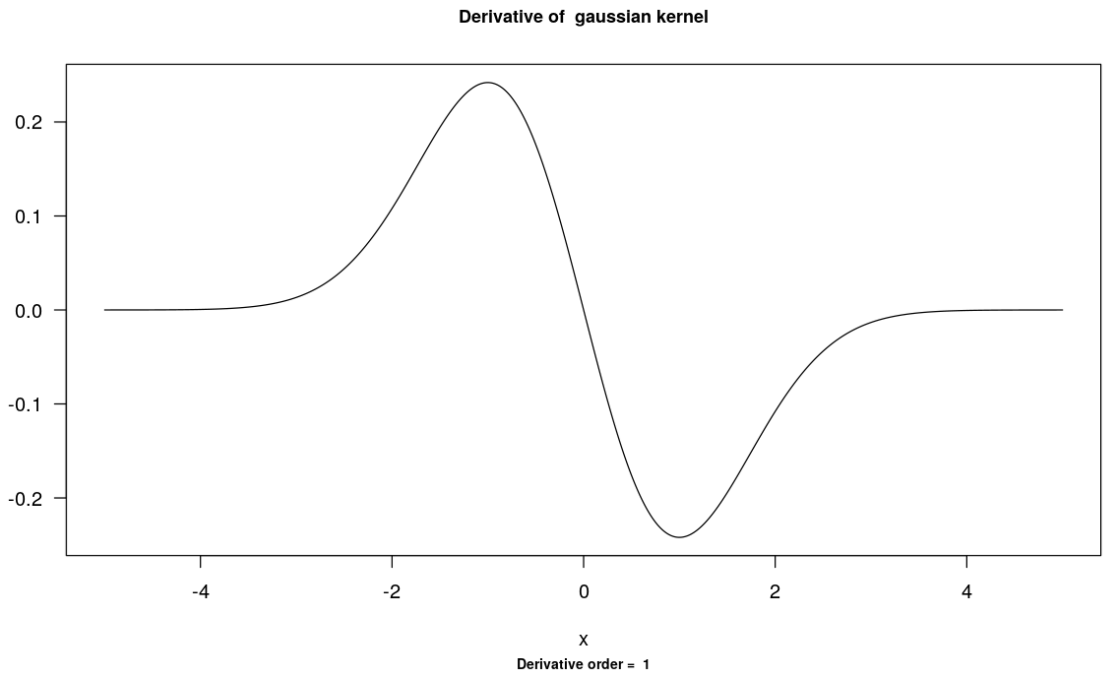
Полная документация с примерами https://cran.r-project.org/web/packages/kedd/vignettes/kedd.pdf (https://cran.r-project.org/web/packages/kedd/vignettes/kedd.pdf)

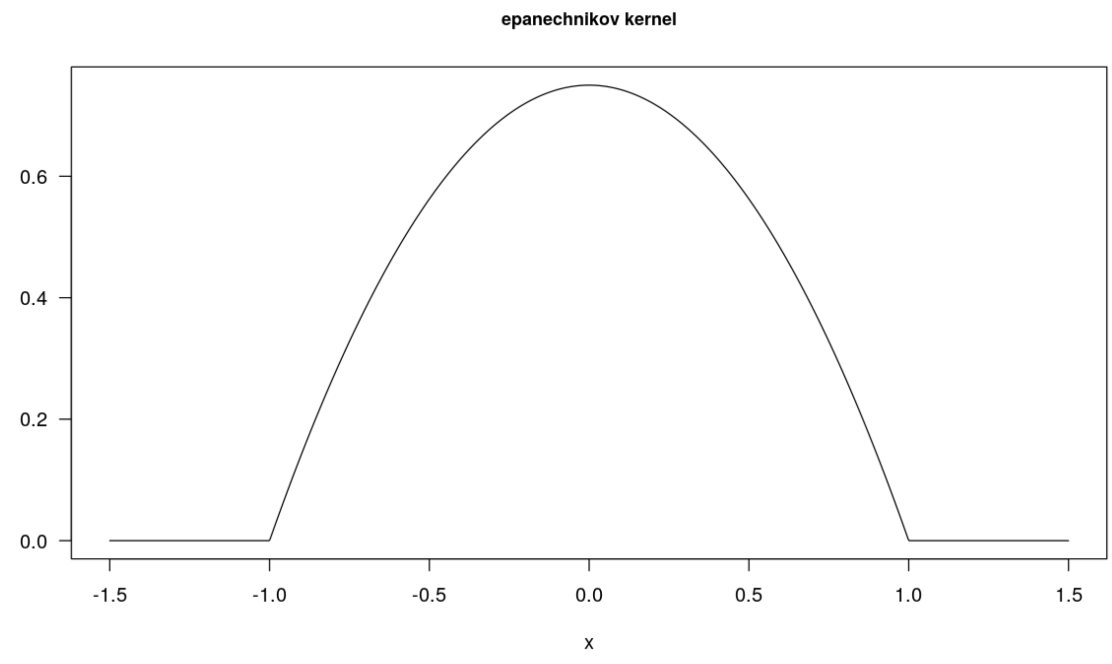
Ядра. Параметр `deriv.order` --- порядок производной, по умолчанию 0.

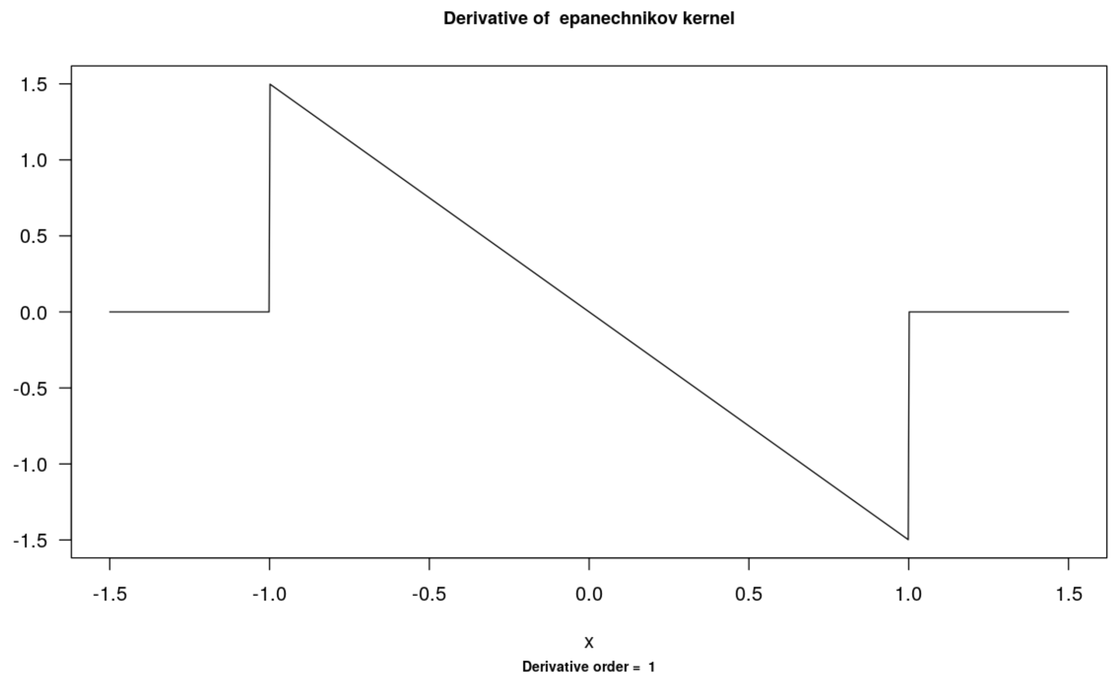
In [4]:

```
plot(kernel.fun(kernel = "gaussian"))
plot(kernel.fun(kernel = "gaussian", deriv.order = 1))
plot(kernel.fun(kernel = "epanechnikov"))
plot(kernel.fun(kernel = "epanechnikov", deriv.order = 1))
```









Сгенерируем выборку

In [5]:

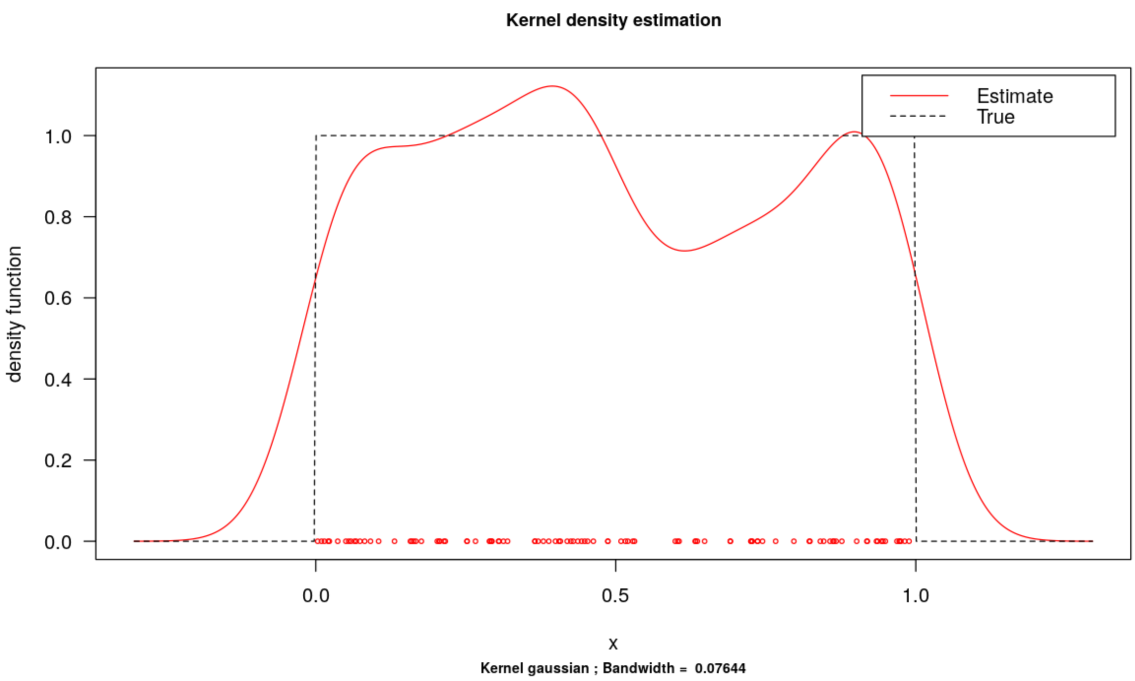
```
sample <- runif(n = 100)
```

Оценка плотности при помощи гауссовского ядра

In [6]:

```r
dens.est <- dkde(sample, deriv.order = 0)
dens.est
plot(dens.est, dunif)
points(sample, rep(0, 100), cex = 0.5, col = 'red')
```

```
Data: sample (100 obs.);        Kernel: gaussian

Derivative order: 0;    Bandwidth 'h' = 0.07644

  eval.points           est.fx
 Min.   :-0.30254   Min.   :0.00006
 1st Qu.: 0.09673   1st Qu.:0.13380
 Median : 0.49600   Median :0.77389
 Mean   : 0.49600   Mean   :0.62492
 3rd Qu.: 0.89527   3rd Qu.:0.97468
 Max.   : 1.29454   Max.   :1.12197
```

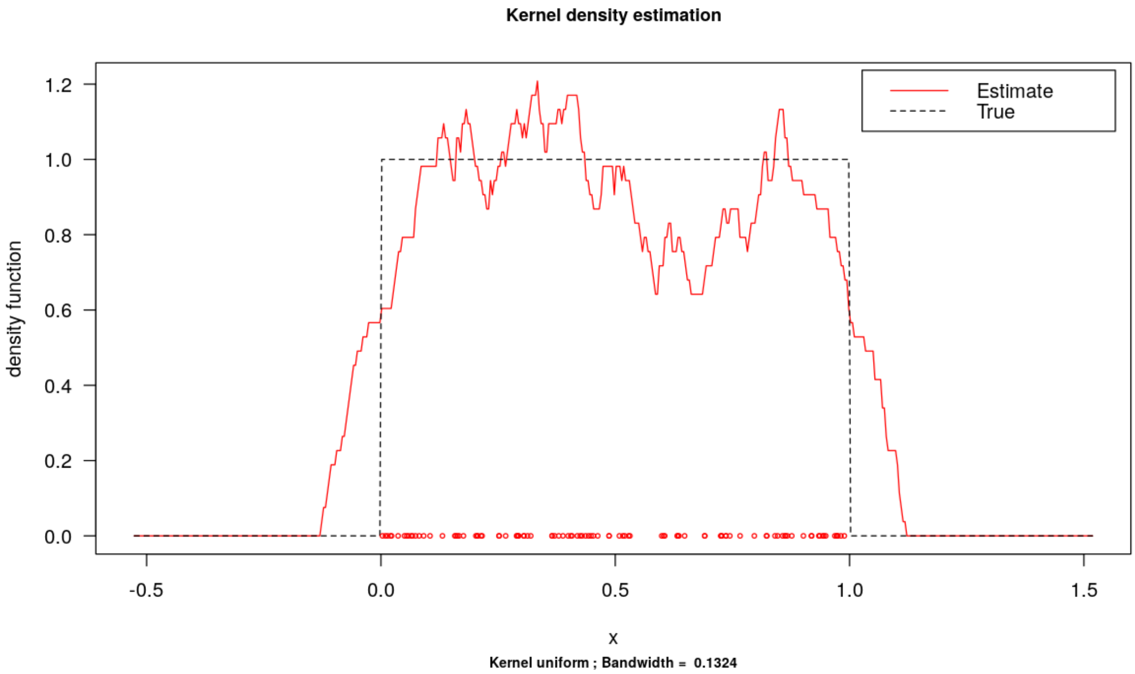


Оценка плотности при помощи равномерного ядра

In [7]:

```
dens.est <- dkde(sample, deriv.order = 0, kernel = 'uniform')
dens.est
plot(dens.est, dunif)
points(sample, rep(0, 100), cex = 0.5, col = 'red')
```

```
Data: sample (100 obs.);        Kernel: uniform

Derivative order: 0;    Bandwidth 'h' = 0.1324

  eval.points           est.fx
 Min.   :-0.52648   Min.   :0.0000
 1st Qu.:-0.01524   1st Qu.:0.0000
 Median : 0.49600   Median :0.5663
 Mean   : 0.49600   Mean   :0.4878
 3rd Qu.: 1.00724   3rd Qu.:0.9061
 Max.   : 1.51848   Max.   :1.2082
```



Один из способов выбора оптимального h (огромные формулы см. в документации)

In [8]:

```
h.mlcv(sample)
h.mlcv(sample, kernel = 'epanechnikov')
```

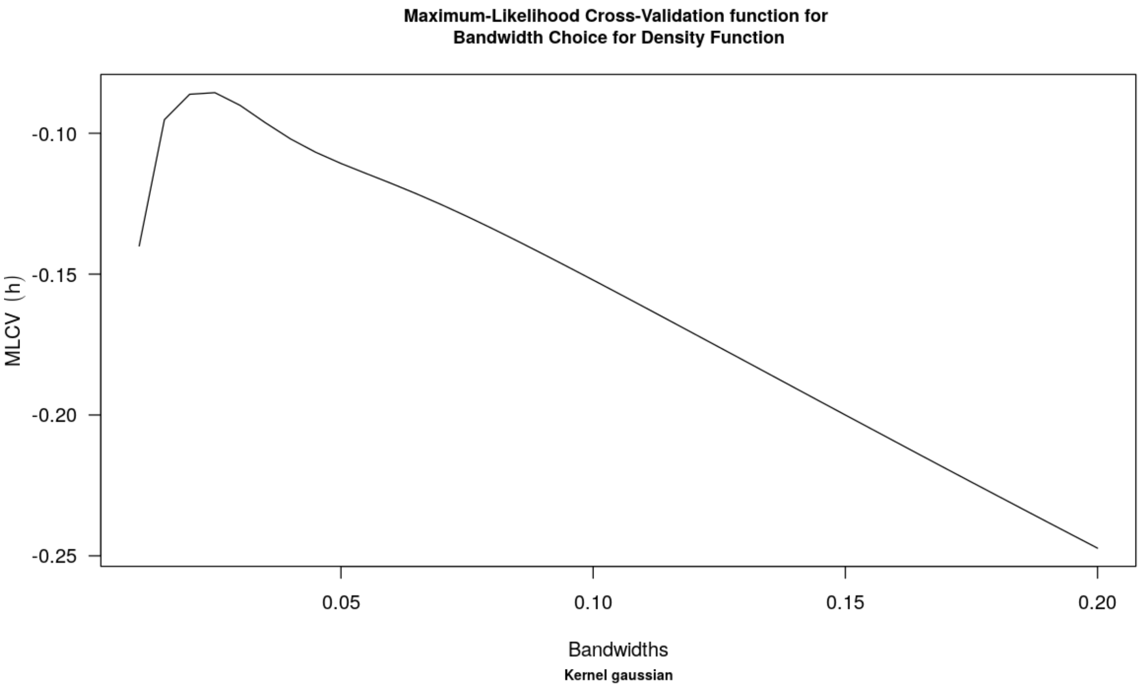
```
Call:           Maximum-Likelihood Cross-Validation

Data: sample (100 obs.);        Kernel: gaussian
Max CV = -0.1576;       Bandwidth 'h' = 0.1058


Call:           Maximum-Likelihood Cross-Validation

Data: sample (100 obs.);        Kernel: epanechnikov
Max CV = -0.1216;       Bandwidth 'h' = 0.1356
```

In [9]:

```r
plot(h.mlcv(sample), seq.bws = seq(0.01, 0.2, 0.005))
```

$kernel
'gaussian'
$seq.bws
0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045 0.05 0.055 0.06 0.065 0.07
0.075 0.08 0.085 0.09 0.095 0.1 0.105 0.11 0.115 0.12 0.125 0.13 0.135
0.14 0.145 0.15 0.155 0.16 0.165 0.17 0.175 0.18 0.185 0.19 0.195 0.2

$mlcv
-0.139998465670239 -0.0951762825593873 -0.0861663944807429
-0.0855960860640676 -0.0901099876601107 -0.0962614914653225
-0.102009706177979 -0.106767848351005 -0.110726564884575 -0.114298406361598
-0.117827863704291 -0.121509460286261 -0.125412059385057 -0.129531035070967
-0.133831094925963 -0.138271299159517 -0.142816018174439 -0.147437793567385
-0.152116584487687 -0.156837928129434 -0.161591155989942 -0.166368042268051
-0.171161911816167 -0.175967110811567 -0.180778722785866 -0.185592431697073
-0.190404462246987 -0.195211553410425 -0.20001094049224 -0.20480033398072
-0.20957789131175 -0.214342181882895 -0.219092147531763 -0.223827061173235
-0.228546486033333 -0.233250237343923 -0.237938347718389 -0.242611036841796
-0.247268685633493



In [10]:

```r
h.ccv(sample)
```

```
Call:           Complete Cross-Validation

Derivative order = 0
Data: sample (100 obs.);        Kernel: gaussian
Min CCV = 0.04574897;   Bandwidth 'h' = 0.1247397
```

In [11]:

```
plot(h.ccv(sample), seq.bws = seq(0.01, 0.4, 0.005))
```

$kernel

'gaussian'

$deriv.order

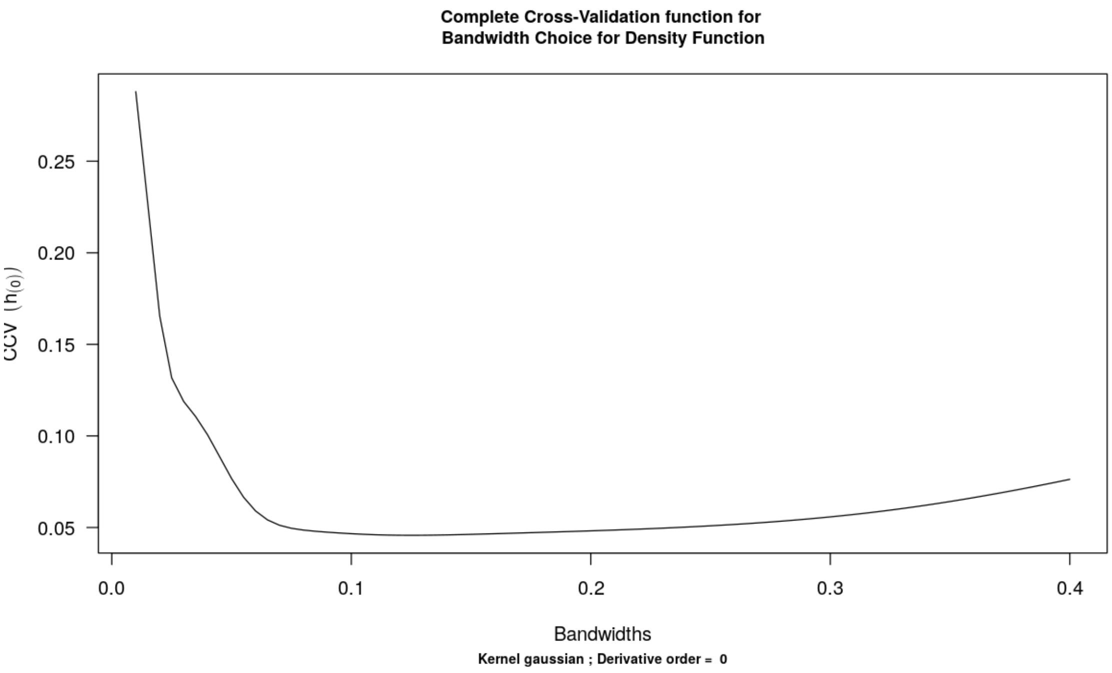
0

$seq.bws

0.01 0.015 0.02 0.025 0.03 0.035 0.04 0.045 0.05 0.055 0.06 0.065 0.07
0.075 0.08 0.085 0.09 0.095 0.1 0.105 0.11 0.115 0.12 0.125 0.13 0.135
0.14 0.145 0.15 0.155 0.16 0.165 0.17 0.175 0.18 0.185 0.19 0.195 0.2
0.205 0.21 0.215 0.22 0.225 0.23 0.235 0.24 0.245 0.25 0.255 0.26 0.265
0.27 0.275 0.28 0.285 0.29 0.295 0.3 0.305 0.31 0.315 0.32 0.325 0.33
0.335 0.34 0.345 0.35 0.355 0.36 0.365 0.37 0.375 0.38 0.385 0.39 0.395
0.4

$ccv

0.287963671095176 0.226936537618265 0.165501445491088 0.131701552675894
0.118816481237368 0.110545054299524 0.100560987609278 0.0886771869569382
0.0767322694699643 0.0665542958311538 0.0590489272336928
0.0541471704524671 0.0512282399515256 0.0495681221305595
0.0485934044155884 0.0479448249932247 0.0474389890643751
0.0470035516820902 0.0466228152642384 0.046302882401415 0.0460532713592703
0.0458792073799733 0.0457797915545902 0.0457489098359937
0.0457770825468383 0.0458533352775356 0.0459666967491676
0.0461072174081811 0.0462665446095359 0.0464381473554183
0.0466172947289328 0.0468008811095584 0.0469871716766173 0.047175521052712
0.0473660998636473 0.047559649857783 0.0477572780730224 0.047960293728348
0.0481700871894285 0.0483880477105234 0.048615515076262 0.0488537593651477
0.0491039826208315 0.0493673361513538 0.0496449474369801
0.0499379511778479 0.0502475198022367 0.0505748897140642
0.0509213806007579 0.0512884061650979 0.0516774756119429
0.0520901860553591 0.0525282066771552 0.0529932559488506
0.0534870735285421 0.0540113885787337 0.0545678862467902
0.0551581739365952 0.0557837488094239 0.0564459677131076
0.0571460204767267 0.0578849072439324 0.0586634202673344
0.0594821303601029 0.0603413780057938 0.0612412689666445
0.0621816741046785 0.0631622330373754 0.0641823611874539
0.0652412597507583 0.0663379280931505 0.0674711780925198
0.0686396499614664 0.0698418291162472 0.0710760636948946
0.0723405823692781 0.073633512139946 0.0749528958470211 0.0762967091737778



Ядерная регрессия

Полная документация с примерами [https://cran.r-project.org/web/packages/np/vignettes/np.pdf (https://cran.r-project.org/web/packages/np/vignettes/np.pdf)](https://cran.r-project.org/web/packages/np/vignettes/np.pdf)

Возьмем датасет о зависимости заработной платы (логарифм) от возраста по данным переписи населения в Канаде 1971 года. Датасет содержит информацию о 205 мужчинах, имеющих одинаковое образование (grade 13).

In [12]:

```
data("cps71")
cps71[1:5,]
```

A data.frame: 5 × 2

logwage	age
<dbl>	<dbl>
11.1563	21
12.8131	22
13.0960	22
11.6952	22
11.5327	22

Линейная модель (параметрическая)

In [13]:

```
model.par <- lm(logwage ~ age + I(age^2), data = cps71)
summary(model.par)
```

```
Call:
lm(formula = logwage ~ age + I(age^2), data = cps71)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.4041 -0.1711  0.0884  0.3182  1.3940 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) 10.0419773  0.4559986  22.022  < 2e-16 ***
age          0.1731310  0.0238317   7.265 7.96e-12 ***
I(age^2)    -0.0019771  0.0002898  -6.822 1.02e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5608 on 202 degrees of freedom
Multiple R-squared:  0.2308,	Adjusted R-squared:  0.2232 
F-statistic:  30.3 on 2 and 202 DF,  p-value: 3.103e-12
```

Нелинейная

In [14]:

```
?npreg
```

In [15]:

```
model.np <- npreg(logwage ~ age, data = cps71, regtype = "ll",
                  bwmethod = "cv.aic", gradients = TRUE)
summary(model.np)
```

```
Regression Data: 205 training points, in 1 variable(s)
                   age
Bandwidth(s): 2.805308

Kernel Regression Estimator: Local-Linear
Bandwidth Type: Fixed
Residual standard error: 0.5215268
R-squared: 0.3251639

Continuous Kernel Type: Second-Order Gaussian
No. Continuous Explanatory Vars.: 1
```

In [16]:

```
npsigtest.res <- npsigtest(model.np)
```

```
Bootstrap replication 1/399 for variable 1 of (1)...Bootstrap replic
ation 2/399 for variable 1 of (1)...Bootstrap replication 3/399 for
variable 1 of (1)...Bootstrap replication 4/399 for variable 1 of
(1)...Bootstrap replication 5/399 for variable 1 of (1)...Bootstrap
replication 6/399 for variable 1 of (1)...Bootstrap replication 7/39
9 for variable 1 of (1)...Bootstrap replication 8/399 for variable 1
of (1)...Bootstrap replication 9/399 for variable 1 of (1)...Bootstr
ap replication 10/399 for variable 1 of (1)..Bootstrap replication 1
1/399 for variable 1 of (1)..Bootstrap replication 12/399 for variab
le 1 of (1)..Bootstrap replication 13/399 for variable 1 of (1)..Boo
tstrap replication 14/399 for variable 1 of (1)..Bootstrap replicati
on 15/399 for variable 1 of (1)..Bootstrap replication 16/399 for va
riable 1 of (1)..Bootstrap replication 17/399 for variable 1 of
(1)..Bootstrap replication 18/399 for variable 1 of (1)..Bootstrap r
eplication 19/399 for variable 1 of (1)..Bootstrap replication 20/39
9 for variable 1 of (1)..Bootstrap replication 21/399 for variable 1
of (1)..Bootstrap replication 22/399 for variable 1 of (1)..Bootstra
p replication 23/399 for variable 1 of (1)..Bootstrap replication 2
4/399 for variable 1 of (1)..Bootstrap replication 25/399 for variab
```

In [17]:

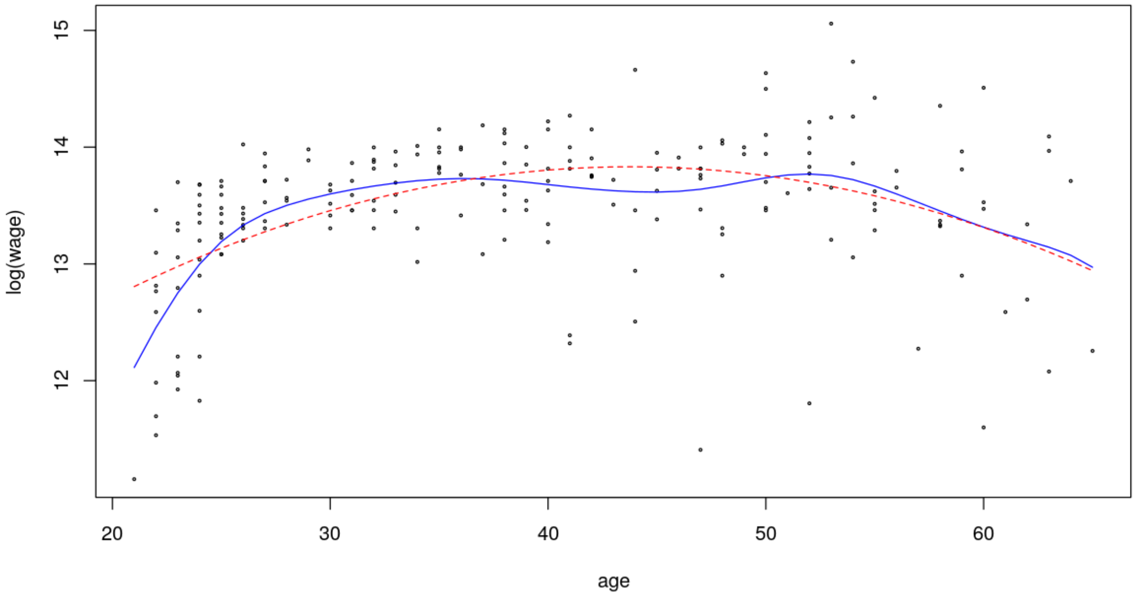
```
npsigtest.res
```

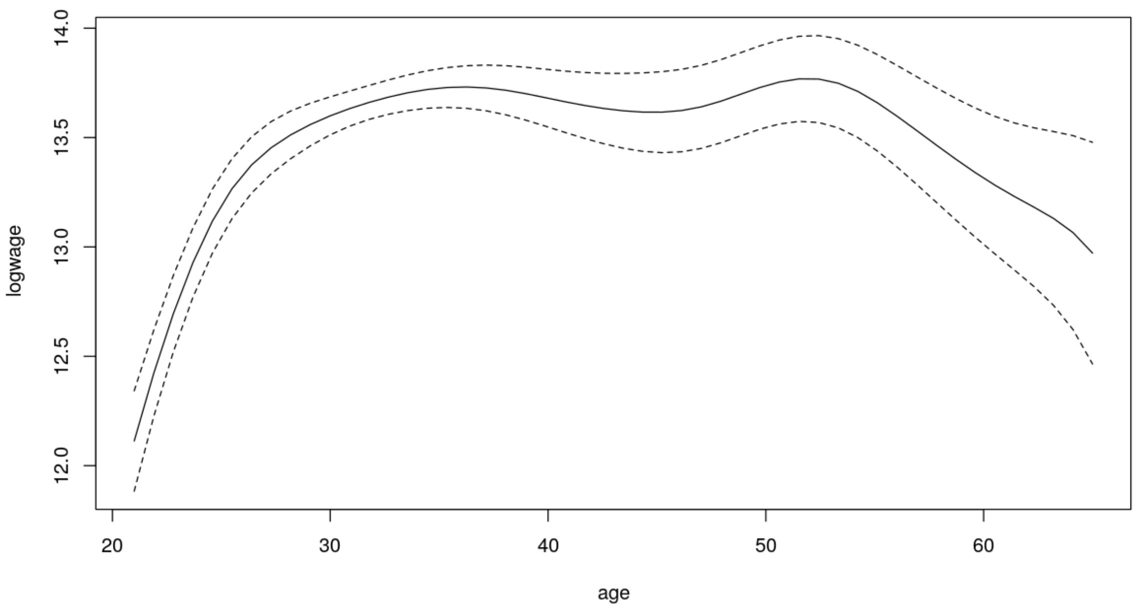
```
Kernel Regression Significance Test
Type I Test with IID Bootstrap (399 replications, Pivot = TRUE, joint
= FALSE)
Explanatory variables tested for significance:
age (1)

                   age
Bandwidth(s): 2.805308

Individual Significance Tests
P Value:
age < 2.22e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

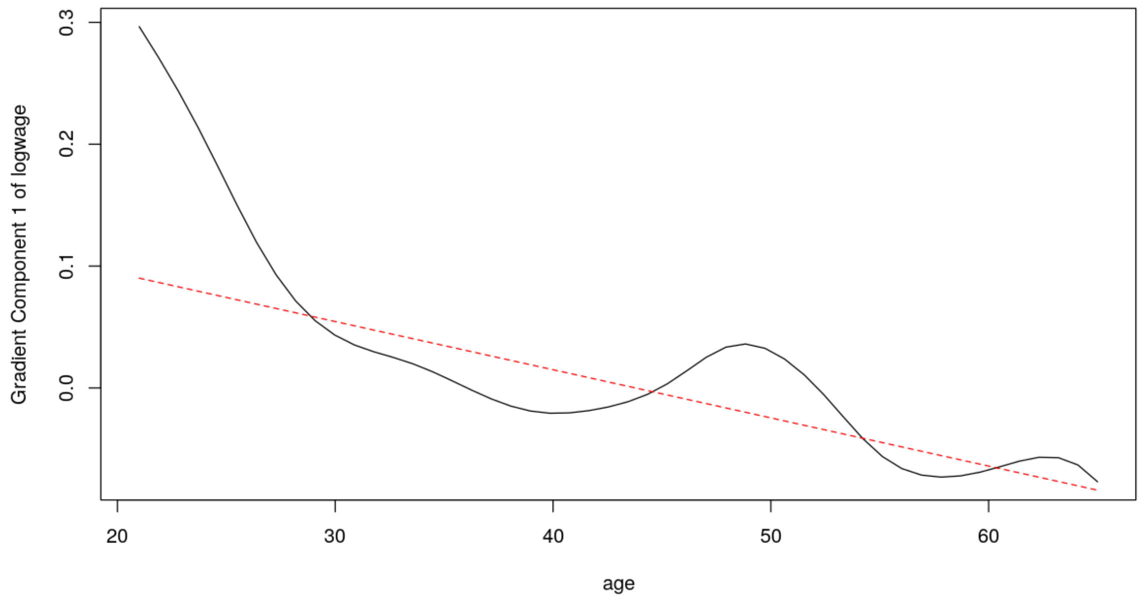
Графики регрессий и их градиента вместе с доверительным интервалом

In [18]:

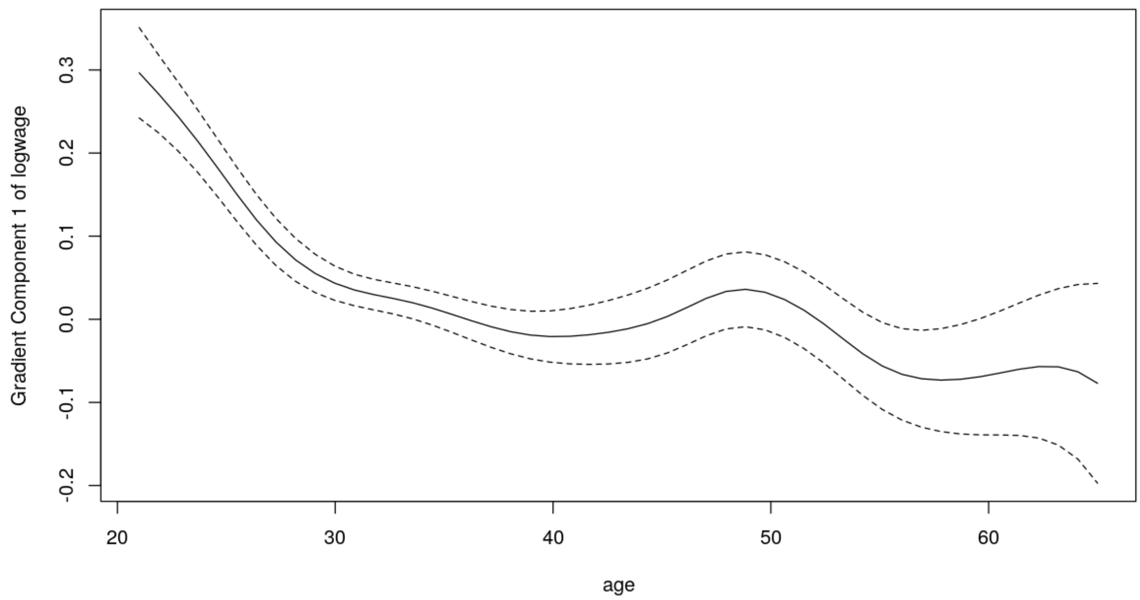
```
plot(cps71$age, cps71$logwage, xlab = "age", ylab = "log(wage)", cex=0.3)
lines(cps71$age, fitted(model.np), lty = 1, col = "blue")
lines(cps71$age, fitted(model.par), lty = 2, col = " red")

plot(model.np, plot.errors.method = "asymptotic")

plot(model.np, gradients = TRUE)
lines(cps71$age, coef(model.par)[2]+2*cps71$age*coef(model.par)[3],
      lty = 2, col = "red")

plot(model.np, gradients = TRUE, plot.errors.method = "asymptotic")
```







Gradient Component 1 of logwage
0.3
0.2
0.1
0.0
20
30
40
50
60
age



Gradient Component 1 of logwage
0.3
0.2
0.1
0.0
-0.1
-0.2
20
30
40
50
60
age