Лабораторная работа

Калганов Максим

Задание 1

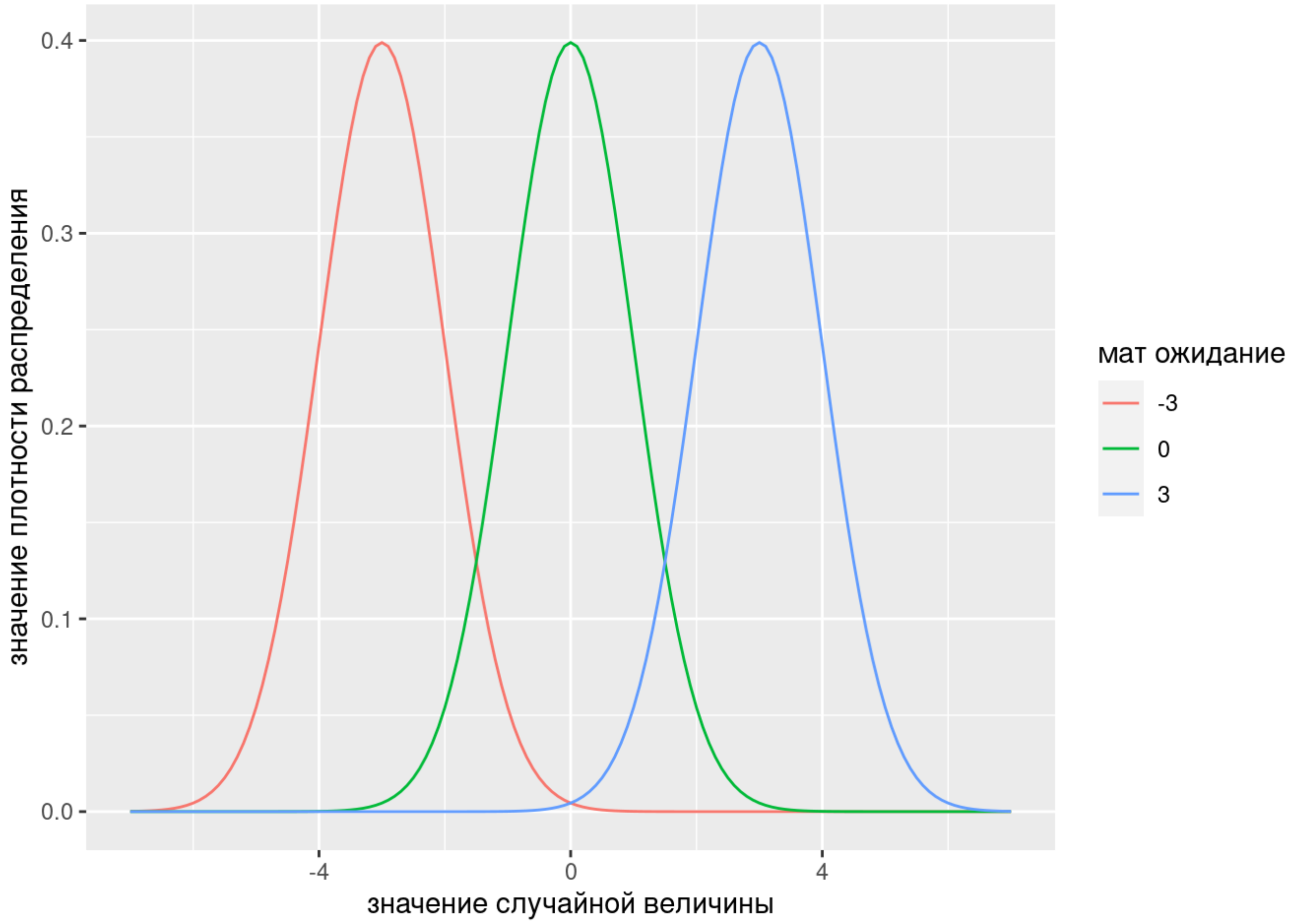
Описание задания 1

1. Рассмотреть 3 распределения, для каждого необходимо построить график плотности распределения (если плотность распределения неизвестна, то по характеристической функции найти плотность), функции распределения, характеристической функции.
2. Провести анализ параметров распределения и графиков плотности распределения (одни параметры изменяются, остальные фиксированые).
3. Вычислить семиинварианты и найти E(x), V(x), S(x) и K(x) через них.
4. Найти квантили: верхний, средний (медиана), нижний и показать как меняются квантили при изменении параметров распределения.

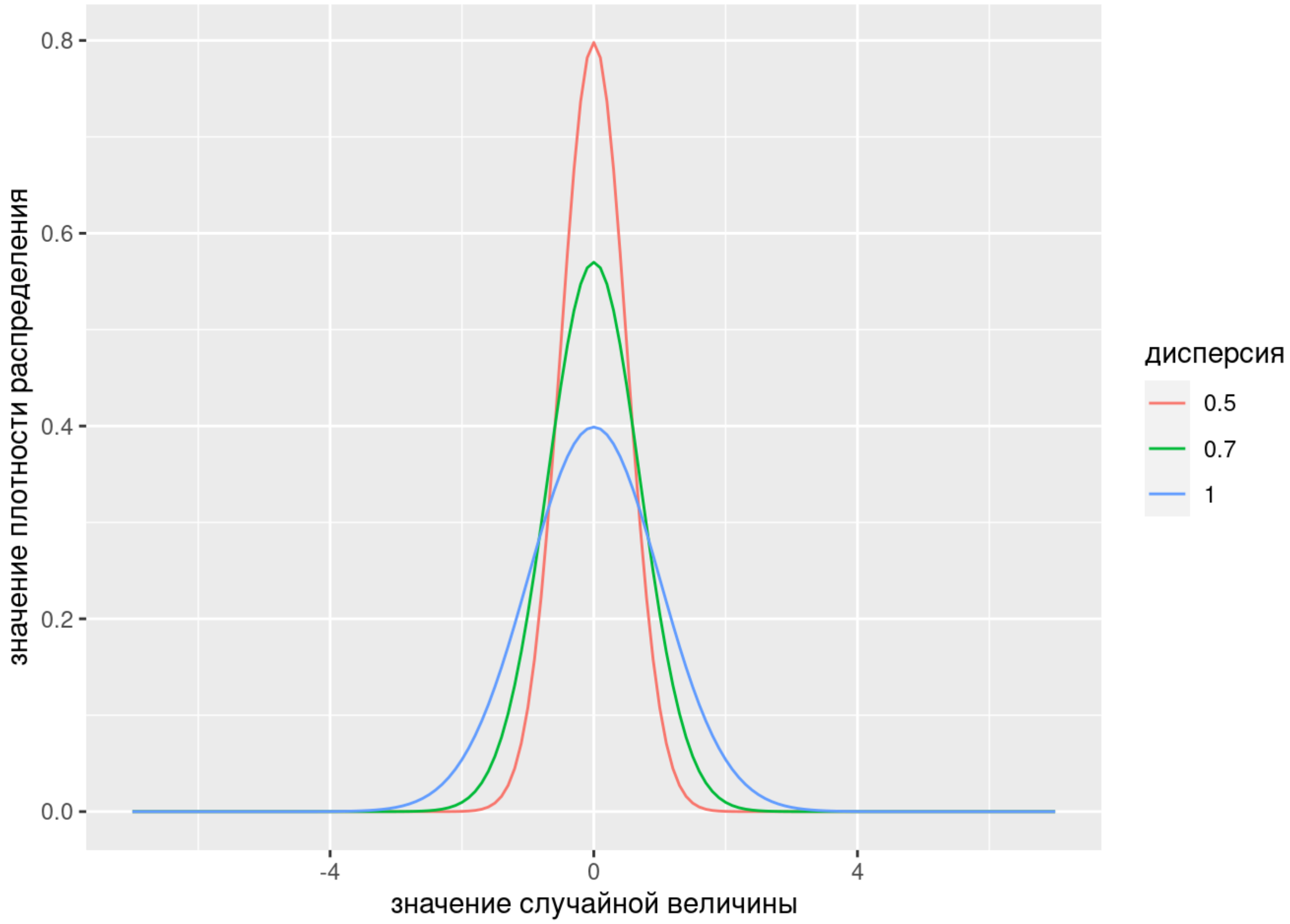
```
library(ggplot2)
library(stabledist)
library(ggforce)
library(SymTS)
library(comprehenr)
library(grid)
library(gridExtra)
library(rmutil)
library(cvar)

set.seed(1234)
x <- seq(-7, 7, 0.1)
```

Нормальное распределение

```
mean1 <- -3
mean2 <- 0
mean3 <- 3
sd1 <- 1
sd2 <- 0.7
sd3 <- 0.5

distr_by_mean <- data.frame(cond = factor(rep(c(mean1, mean2, mean3), each=length
(x))),
                     rating = c(dnorm(x, mean=mean1, sd=sd1),
                                dnorm(x, mean=mean2, sd=sd1),
                                dnorm(x, mean=mean3, sd=sd1)), x=x)

ggplot(distr_by_mean, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "мат ожидание") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```




```
distr_by_sd <- data.frame(cond = factor(rep(c(sd1, sd2, sd3), each=length(x))),
                            rating = c(dnorm(x, mean=mean2, sd=sd1),
                                       dnorm(x, mean=mean2, sd=sd2),
                                       dnorm(x, mean=mean2, sd=sd3)), x=x)

ggplot(distr_by_sd, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "дисперсия") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```



Выводы:

1. Изменение мат ожидания смещает график плотности распределения
2. Изменение дисперсии сжимает, либо вытягивает график плотности распределения

```
print_quant_norm <- function(mean, sd){
  x = c(0.25, 0.5, 0.75)
  print(paste0("мат ожидание = ", mean))
  print(paste0("дисперсия = ", sd))
  quantiles <- paste(qnorm(x, mean=mean, sd=sd), collapse = ", ")
  print(paste0("квантили = ", quantiles))
}
print_quant_norm(mean1, sd1)
```

```
## [1] "мат ожидание = -3"
## [1] "дисперсия = 1"
## [1] "квантили = -3.67448975019608, -3, -2.32551024980392"
```

```
print_quant_norm(mean2, sd1)
```

```
## [1] "мат ожидание = 0"
## [1] "дисперсия = 1"
## [1] "квантили = -0.674489750196082, 0, 0.674489750196082"
```

```
print_quant_norm(mean3, sd1)
```

```
## [1] "мат ожидание = 3"
## [1] "дисперсия = 1"
## [1] "квантили = 2.32551024980392, 3, 3.67448975019608"
```

```
print_quant_norm(mean1, sd1)
```

```
## [1] "мат ожидание = -3"
## [1] "дисперсия = 1"
## [1] "квантили = -3.67448975019608, -3, -2.32551024980392"
```

```
print_quant_norm(mean1, sd2)
```

```
## [1] "мат ожидание = -3"
## [1] "дисперсия = 0.7"
## [1] "квантили = -3.47214282513726, -3, -2.52785717486274"
```

```
print_quant_norm(mean1, sd3)
```

```
## [1] "мат ожидание = -3"
## [1] "дисперсия = 0.5"
## [1] "квантили = -3.33724487509804, -3, -2.66275512490196"
```

1. $X \sim \mathcal{N}(\mu, \sigma^2)$

$$C_n(x) = i^{-n} \left. \frac{d^n \ln \varphi_x(t)}{dt^n} \right|_{t=0}$$

$$\varphi_x(t) = e^{it\mu - \frac{\sigma^2 t^2}{2}}, \quad \ln \varphi_x(t) = it\mu - \frac{\sigma^2 t^2}{2}$$

$$C_1(x) = \left. \frac{1}{i}(i\mu - \sigma^2 t) \right|_{t=0} = \mu$$

$$C_2(x) = \left. \frac{1}{-1}(-\sigma^2) \right|_{t=0} = \sigma^2$$

$$C_3(x) = C_4(x) = 0$$

$$E(x) = C_1(x) = \mu$$

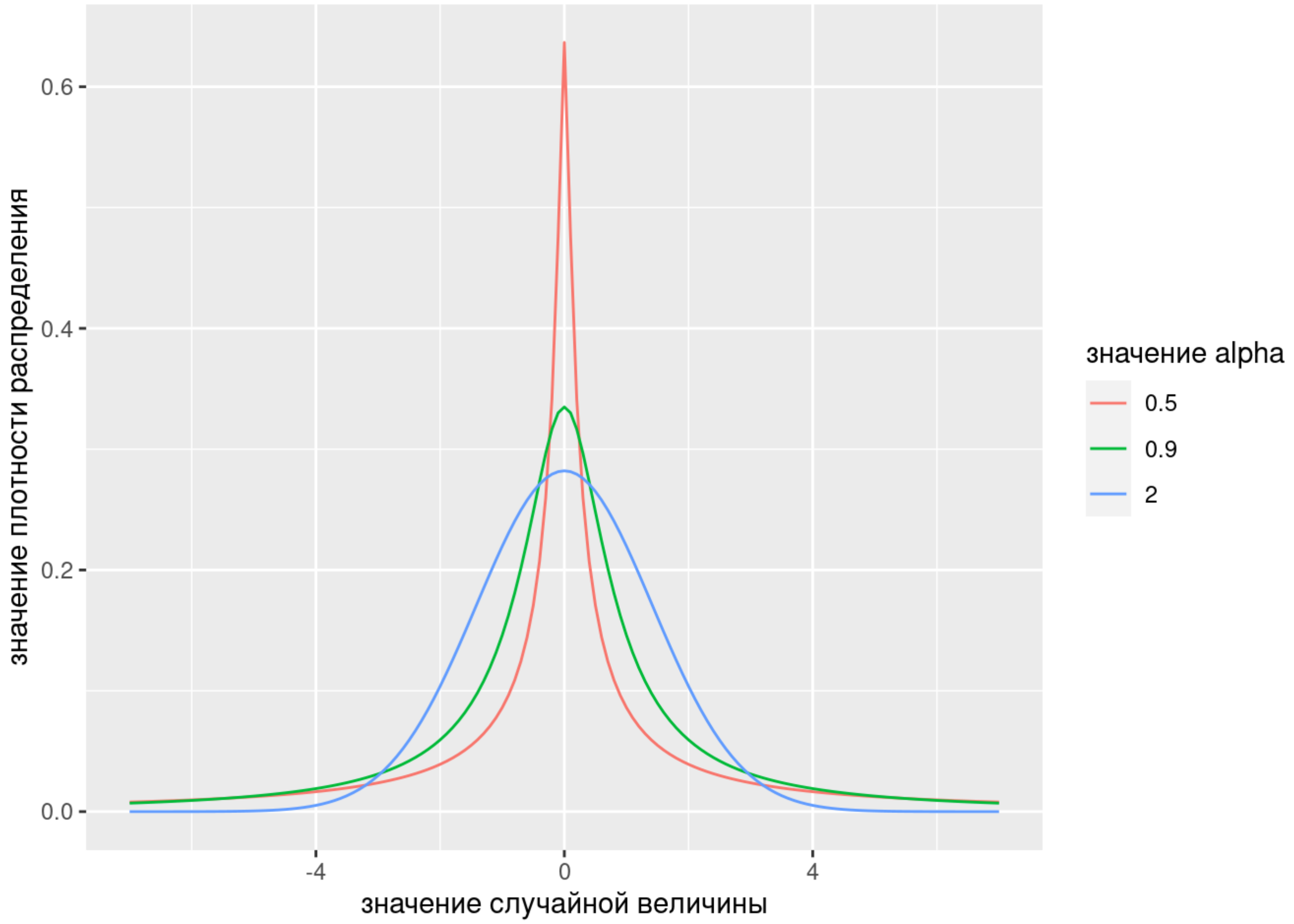
$$V(x) = C_2(x) = \sigma^2$$

$$S(x) = \frac{C_3(x)}{C_2(x)^{3/2}} = \frac{0}{(\sigma^2)^{3/2}} = 0$$

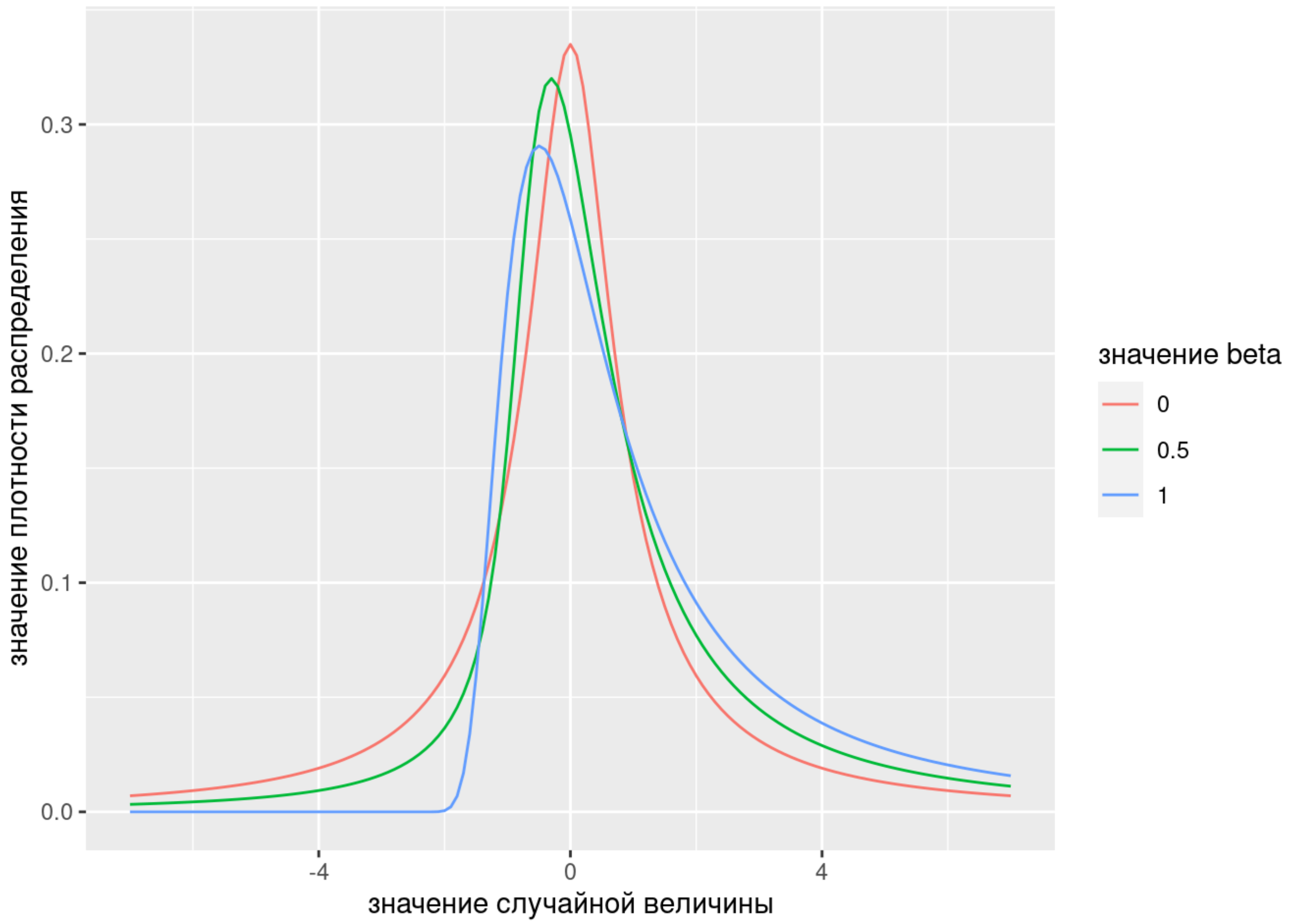
$$K(x) = \frac{C_4(x)}{(C_2(x))^2} = \frac{0}{(\sigma^2)^2} = 0$$

Устойчивое распределение

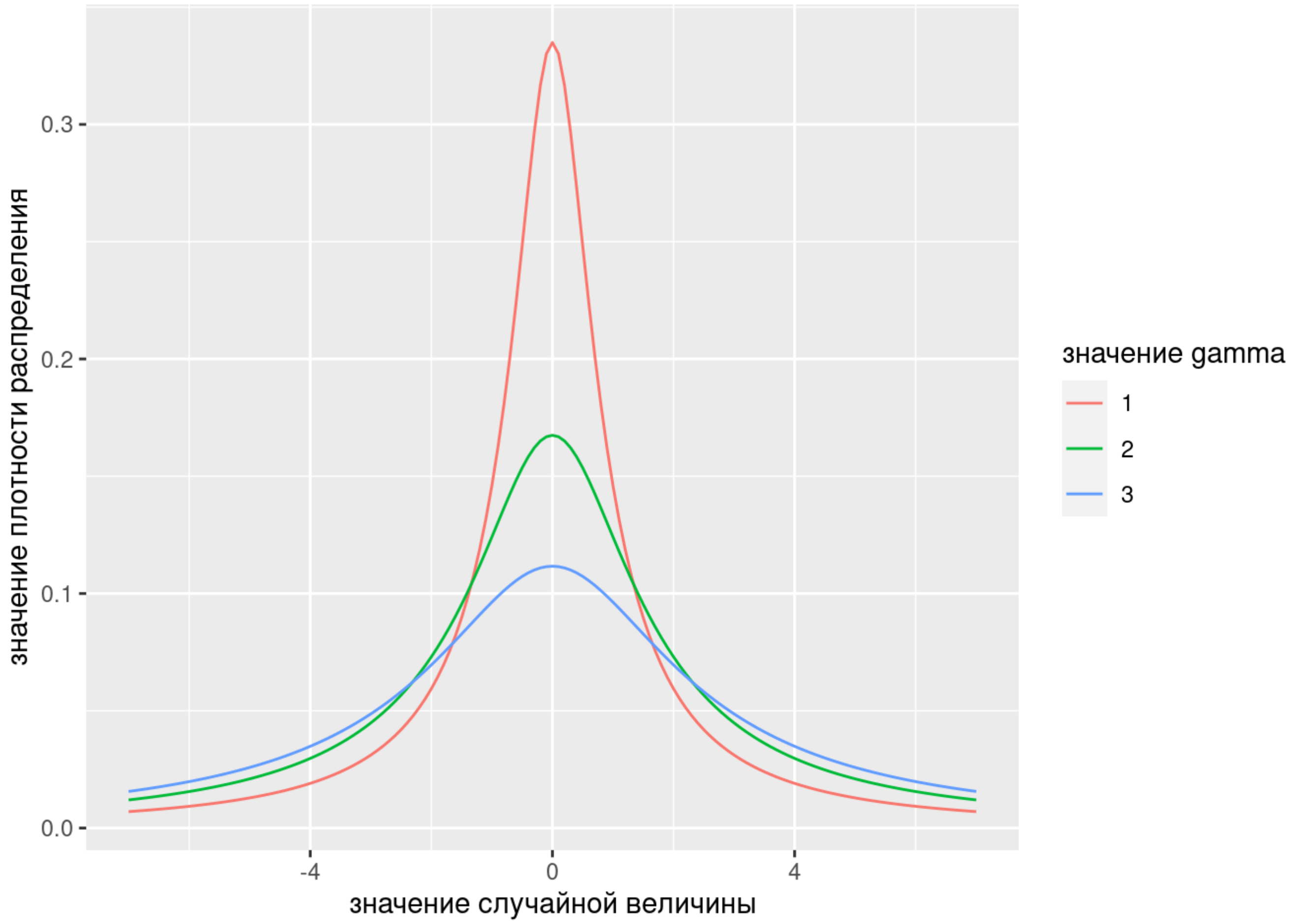
```
alpha1 <- 0.5
alpha2 <- 0.9
alpha3 <- 2

beta1 <- 0.5
beta2 <- 0
beta3 <- 1

gamma1 <- 2
gamma2 <- 1
gamma3 <- 3

delta1 <- -1
delta2 <- 0
delta3 <- 1


distr_by_alpha <- data.frame(cond = factor(rep(c(alpha1, alpha2, alpha3), each=length
(x))),
                             x=x,
                            rating = c(dstable(x, alpha=alpha1, beta=beta2, gamma=gam
ma2, delta = delta2),
                                       dstable(x, alpha=alpha2, beta=beta2, gamma=gam
ma2, delta = delta2),
                                       dstable(x, alpha=alpha3, beta=beta2, gamma=gam
ma2, delta = delta2)))

ggplot(distr_by_alpha, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение alpha") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```



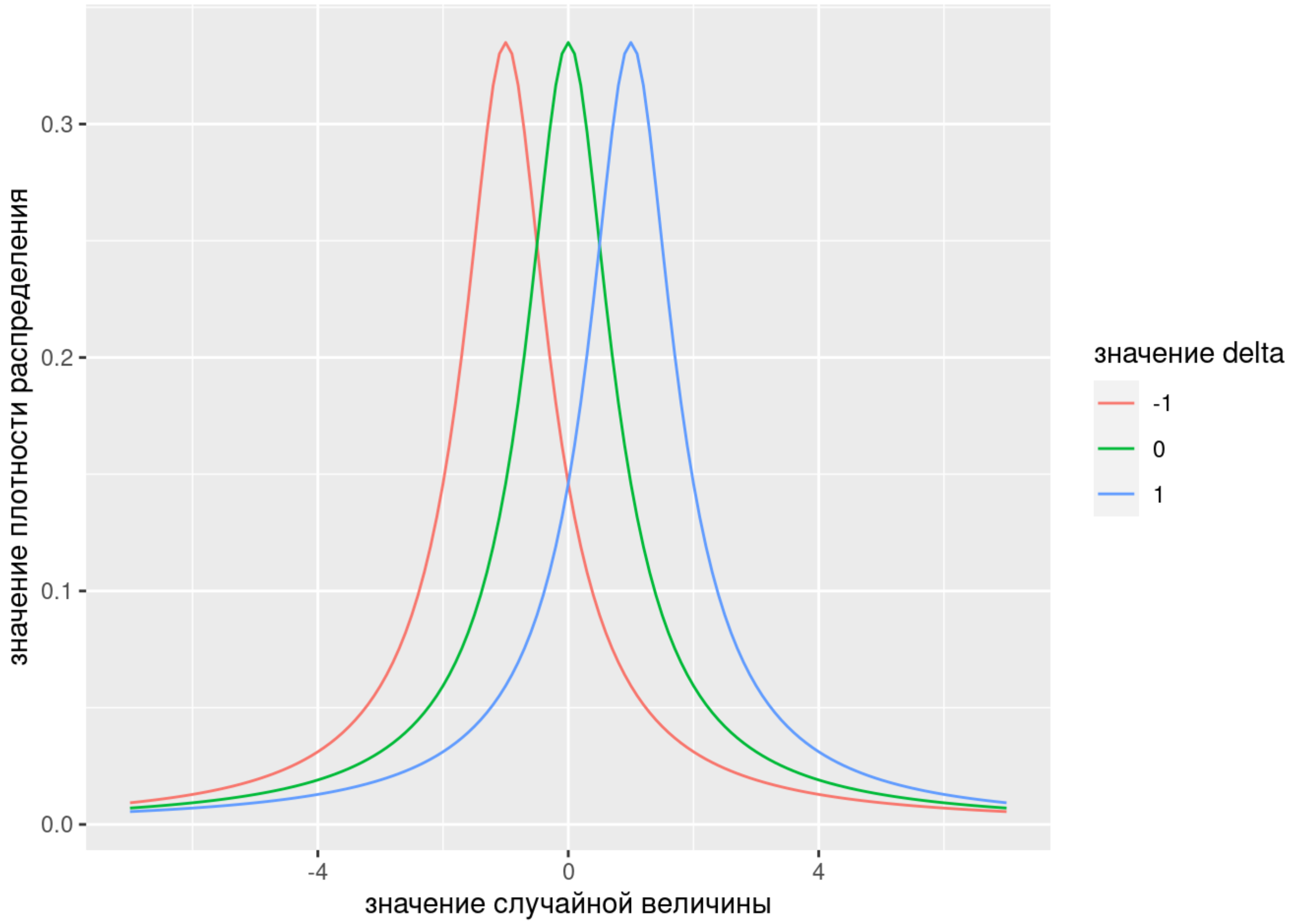

```
distr_by_beta <- data.frame(cond = factor(rep(c(beta1, beta2, beta3), each=length
(x))),
                            x=x,
                             rating = c(dstable(x, alpha=alpha2, beta=beta1, gamma=ga
mma2, delta = delta2),
                                        dstable(x, alpha=alpha2, beta=beta2, gamma=ga
mma2, delta = delta2),
                                        dstable(x, alpha=alpha2, beta=beta3, gamma=ga
mma2, delta = delta2)))

ggplot(distr_by_beta, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение beta") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```




```
distr_by_gamma <- data.frame(cond = factor(rep(c(gamma1, gamma2, gamma3), each=length
(x))),
                             x=x,
                             rating = c(dstable(x, alpha=alpha2, beta=beta2, gamma=ga
mma1, delta = delta2),
                                        dstable(x, alpha=alpha2, beta=beta2, gamma=ga
mma2, delta = delta2),
                                        dstable(x, alpha=alpha2, beta=beta2, gamma=ga
mma3, delta = delta2)))

ggplot(distr_by_gamma, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение gamma") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```




```
distr_by_delta <- data.frame(cond = factor(rep(c(delta1, delta2, delta3), each=length
(x))),
                             x=x,
                             rating = c(dstable(x, alpha=alpha2, beta=beta2, gamma=ga
mma2, delta = delta1),
                                        dstable(x, alpha=alpha2, beta=beta2, gamma=ga
mma2, delta = delta2),
                                        dstable(x, alpha=alpha2, beta=beta2, gamma=ga
mma2, delta = delta3)))

ggplot(distr_by_delta, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение delta") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```



Выводы:

1. Параметр alpha имеет схожий эффект, что и дисперсия, но судя по формуле характеристической функции имеет более сложное влияние
2. Изменение параметра beta влияет на ассиметрию графика плотности распределения
3. Судя по графикам, параметр gamma отвечает за масштаб
4. Параметр delta имеет схожий эффект, что и мат ожидание

```
print_quant_stable <- function(alpha, beta, gamma, delta){
  x = c(0.25, 0.5, 0.75)
  print(paste0("параметр alpha = ", alpha))
  print(paste0("параметр beta = ", beta))
  print(paste0("параметр gamma = ", gamma))
  print(paste0("параметр delta = ", delta))
  quantiles <- paste(qstable(x, alpha=alpha, beta=beta, gamma=gamma, delta = delta),
 collapse = ", ")
  print(paste0("квантили = ", quantiles))
}

print_quant_stable(alpha1, beta2, gamma2, delta2)
```

```
## [1] "параметр alpha = 0.5"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -1.2838329219335, -1e-09, 1.2838329219335"
```

```
print_quant_stable(alpha2, beta2, gamma2, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -1.01758385350076, -1e-09, 1.01758385350076"
```

```
print_quant_stable(alpha3, beta2, gamma2, delta2)
```

```
## [1] "параметр alpha = 2"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -0.95387255240894, 0, 0.95387255240894"
```

```
print_quant_stable(alpha2, beta1, gamma2, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0.5"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -0.593044365717121, 0.237583627820147, 1.86295129133498"
```

```
print_quant_stable(alpha2, beta2, gamma2, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -1.01758385350076, -1e-09, 1.01758385350076"
```

```
print_quant_stable(alpha2, beta3, gamma2, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 1"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -0.380182808457033, 0.652493541760843, 2.96975428636576"
```

```
print_quant_stable(alpha2, beta2, gamma1, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 2"
## [1] "параметр delta = 0"
## [1] "квантили = -2.03516770700152, -2e-09, 2.03516770700152"
```

```
print_quant_stable(alpha2, beta2, gamma2, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -1.01758385350076, -1e-09, 1.01758385350076"
```

```
print_quant_stable(alpha2, beta2, gamma3, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 3"
## [1] "параметр delta = 0"
## [1] "квантили = -3.05275156050228, -3e-09, 3.05275156050228"
```

```
print_quant_stable(alpha2, beta2, gamma2, delta1)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = -1"
## [1] "квантили = -2.01758385350076, -1.000000001, 0.0175838535007597"
```

```
print_quant_stable(alpha2, beta2, gamma2, delta2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 0"
## [1] "квантили = -1.01758385350076, -1e-09, 1.01758385350076"
```

```
print_quant_stable(alpha2, beta2, gamma2, delta3)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр beta = 0"
## [1] "параметр gamma = 1"
## [1] "параметр delta = 1"
## [1] "квантили = -0.0175838535007597, 0.999999999, 2.01758385350076"
```

2. $x \sim S_\alpha(\sigma, \beta, \mu)$, $\alpha \neq 1, \frac{3}{2}$ (по условию задачи)

$$\ln \varphi_x(t) = i\mu t - \sigma^\alpha |t|^\alpha + i\sigma^\alpha |t|^\alpha \beta \, tg \frac{\alpha\pi}{2} =$$

$$= i\mu t + \sigma^\alpha |t|^\alpha \left(-1 + i\beta \, tg \frac{\alpha\pi}{2}\right)$$

$$c_1(x) = \frac{1}{i}\left(i\mu + \alpha \cdot \sigma^\alpha t^{\alpha-1}\left(-1 + i\beta \, tg \frac{\alpha\pi}{2}\right)\right)\Big|_{t=0} = \mu$$

$$c_2(x) = \frac{1}{-1}\left(\alpha \cdot \sigma^\alpha (\alpha-1) \cdot t^{\alpha-2}\left(-1 + i\beta \, tg \frac{\alpha\pi}{2}\right)\right)\Big|_{t=0} = \begin{cases} 0, & \alpha \neq 2 \\ 2\sigma^2, & \alpha = 2 \end{cases}$$

$$c_3(x) = c_4(x) = 0$$

$$E(x) = c_1(x) = \mu$$

$$V(x) = c_2(x) = \begin{cases} 0, & \alpha \neq 2 \\ 2\sigma^2, & \alpha = 2 \end{cases}$$

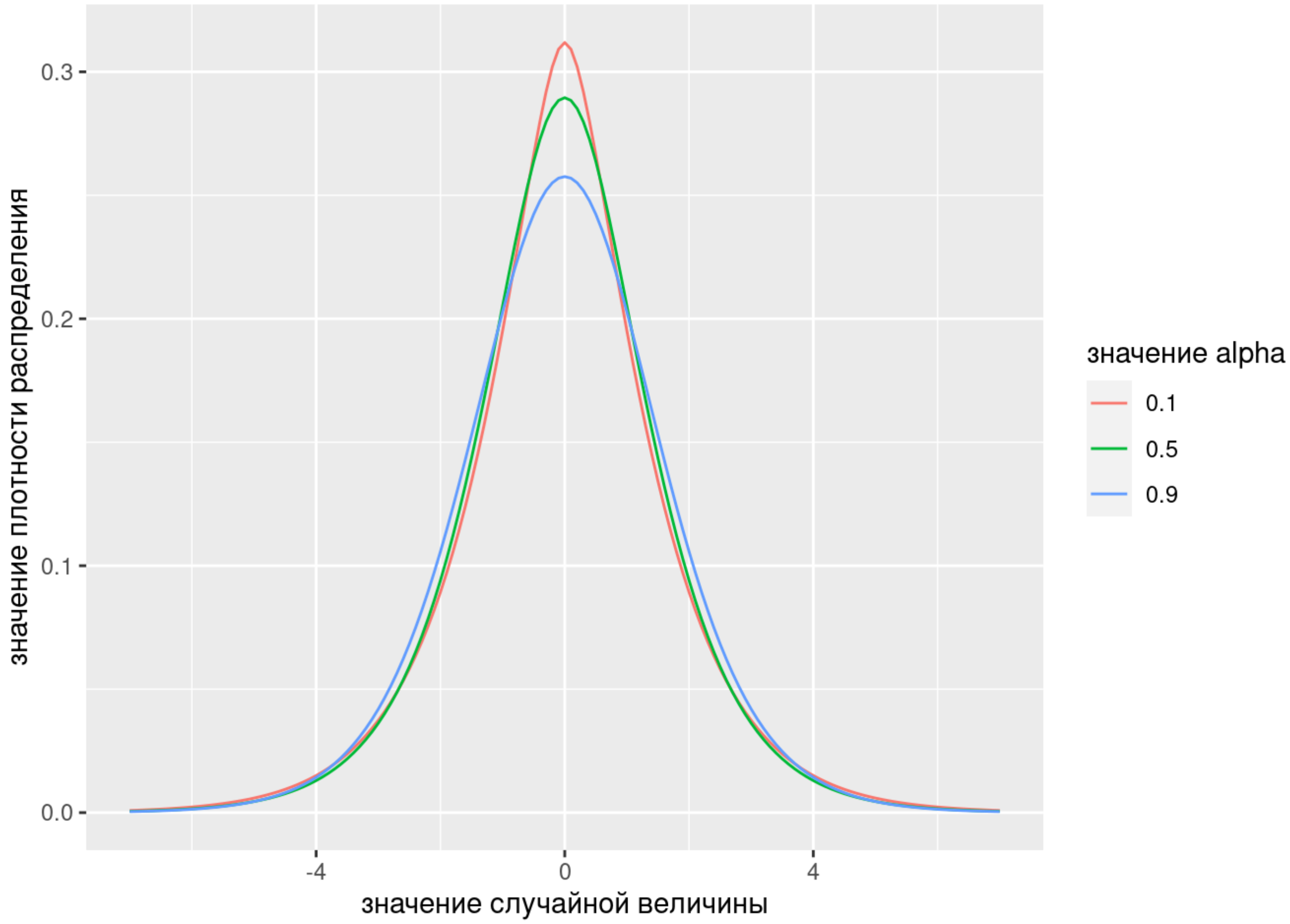
$$S(x) = \frac{c_3(x)}{c_2(x)^{3/2}} = \frac{0}{c_2^{3/2}} = \begin{cases} \text{неопределенность}, & \alpha \neq 2 \\ 0, & \alpha = 2 \end{cases}$$

$$K(x) = \frac{c_4(x)}{c_2(x)^2} = \frac{0}{c_2(x)^2} = \begin{cases} \text{неопределенность}, & \alpha \neq 2 \\ 0, & \alpha = 2 \end{cases}$$

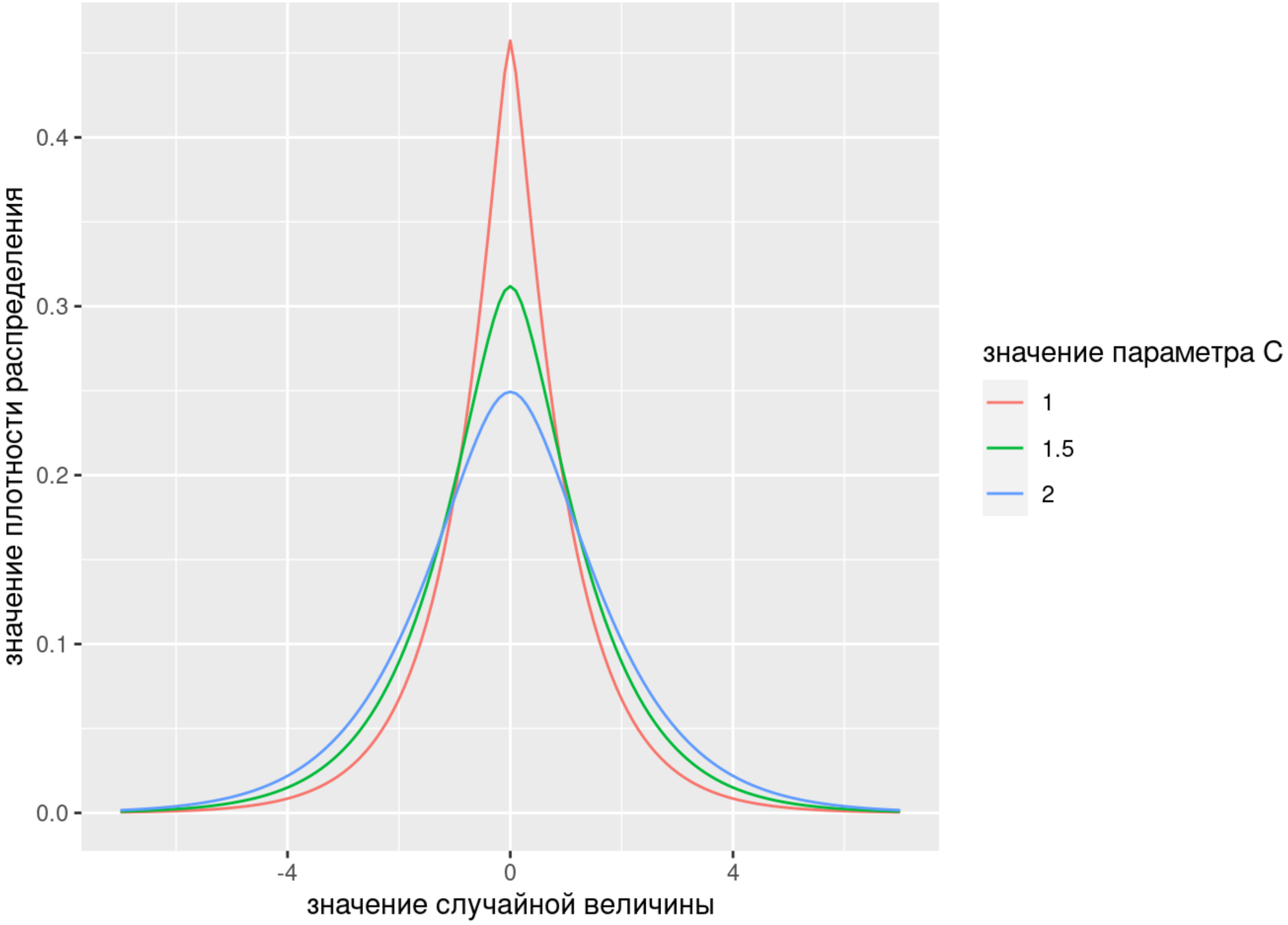
Из лекции следует, что beta параметр является параметром ассиметрами (это же следует из графиков), но по формулам выходит, что S(x) = 0

CTS распределение

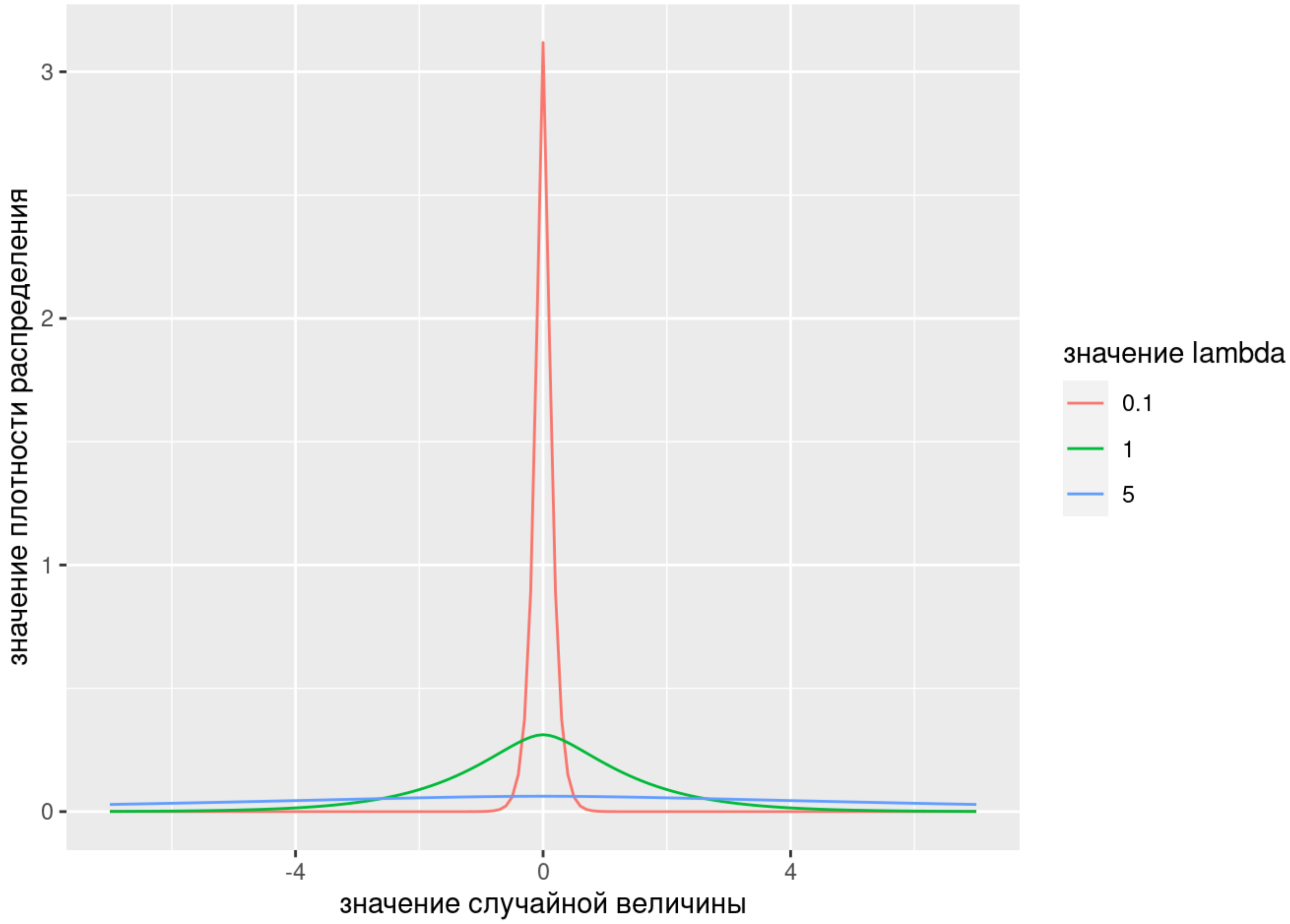
```
cts_alpha1 <- 0.1
cts_alpha2 <- 0.5
cts_alpha3 <- 0.9

lambda1 <- 0.1
lambda2 <- 1
lambda3 <- 5

c1 <- 1
c2 <- 1.5
c3 <- 2

mu1 <- -1
mu2 <- 0
mu3 <- 1

cts_distr_by_cts_alpha <- data.frame(cond = factor(rep(c(cts_alpha1, cts_alpha2, cts_
alpha3), each=length(x))),
                                 x=x,
                                 rating = c(dCTS(x, alpha=cts_alpha1, c=c2, ell=lambd
a2, mu = mu2),
                                        dCTS(x, alpha=cts_alpha2, c=c2, ell=lambda2,
 mu = mu2),
                                        dCTS(x, alpha=cts_alpha3, c=c2, ell=lambda2,
 mu = mu2)))

ggplot(cts_distr_by_cts_alpha, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение alpha") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```



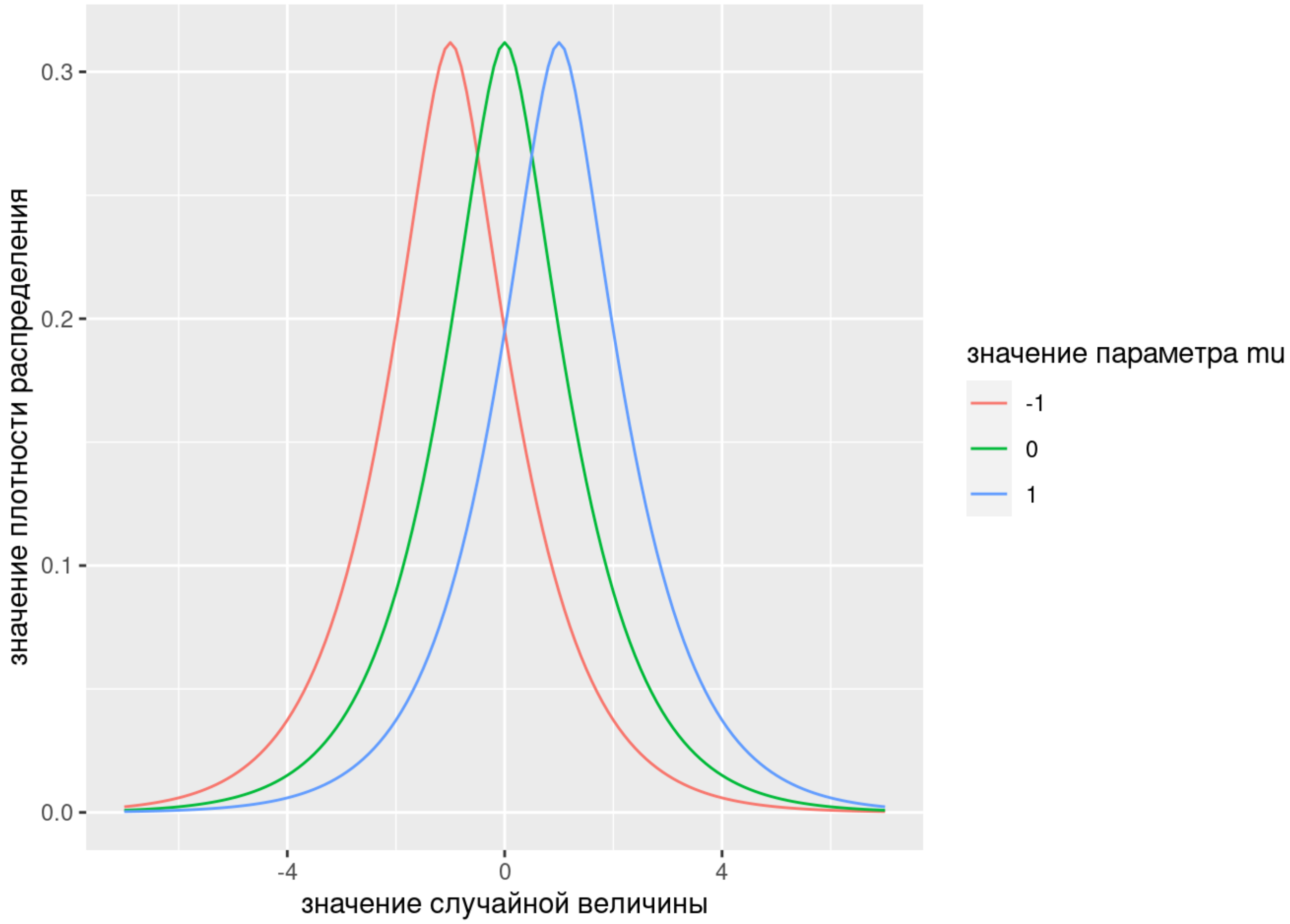

```
cts_distr_by_c <- data.frame(cond = factor(rep(c(c1, c2, c3), each=length(x))),
                             x=x,
                             rating = c(dCTS(x, alpha=cts_alpha1, c=c1, ell=lambda2,
mu = mu2),
                                        dCTS(x, alpha=cts_alpha1, c=c2, ell=lambda2,
mu = mu2),
                                        dCTS(x, alpha=cts_alpha1, c=c3, ell=lambda2,
mu = mu2)))

ggplot(cts_distr_by_c, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение параметра C") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```




```
cts_distr_by_lambda <- data.frame(cond = factor(rep(c(lambda1, lambda2, lambda3), eac
h=length(x))),
                                  x=x,
                                  rating = c(dCTS(x, alpha=cts_alpha1, c=c2, ell=lamb
da1, mu = mu2),
                                             dCTS(x, alpha=cts_alpha1, c=c2, ell=lamb
da2, mu = mu2),
                                             dCTS(x, alpha=cts_alpha1, c=c2, ell=lamb
da3, mu = mu2)))

ggplot(cts_distr_by_lambda, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение lambda") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```




```
cts_distr_by_mu <- data.frame(cond = factor(rep(c(mu1, mu2, mu3), each=length(x))),
                              x=x,
                              rating = c(dCTS(x, alpha=cts_alpha1, c=c2, ell=lambda2,
mu = mu1),
                                         dCTS(x, alpha=cts_alpha1, c=c2, ell=lambda2,
mu = mu2),
                                         dCTS(x, alpha=cts_alpha1, c=c2, ell=lambda2,
mu = mu3)))

ggplot(cts_distr_by_mu, aes(x=x, y=rating, color = cond)) +
  geom_line() +
  labs(col = "значение параметра mu") +
  xlab("значение случайной величины") +
  ylab("значение плотности распределения")
```



Выводы:

Из графиков можно сделать вывод только о том, что все параметры кроме mu имеют схожий эффект с дисперсией. Параметр mu отвечает за сдвиг графика плотности распределения.

```
print_quant_cts <- function(cts_alpha, c, lambda, mu){
  x = c(0.25, 0.5, 0.75)
  print(paste0("параметр alpha = ", cts_alpha))
  print(paste0("параметр C = ", c))
  print(paste0("параметр lambda = ", lambda))
  print(paste0("параметр mu = ", mu))
  quantiles <- paste(qCTS(x, alpha=cts_alpha, c=c, ell=lambda, mu = mu), collapse =
", ")
  print(paste0("квантили = ", quantiles))
}

print_quant_cts(cts_alpha1, lambda2, c2, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -1.0441887374331, 0, 1.0441887374331"
```

```
print_quant_cts(cts_alpha2, lambda2, c2, mu2)
```

```
## [1] "параметр alpha = 0.5"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -1.10882178029597, 0, 1.10882178029597"
```

```
print_quant_cts(cts_alpha3, lambda2, c2, mu2)
```

```
## [1] "параметр alpha = 0.9"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -1.25788637960172, 0, 1.25788637960172"
```

```
print_quant_cts(cts_alpha1, lambda1, c2, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 0.1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -0.04421845704661, 0, 0.04421845704661"
```

```
print_quant_cts(cts_alpha1, lambda2, c2, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -1.0441887374331, 0, 1.0441887374331"
```

```
print_quant_cts(cts_alpha1, lambda3, c2, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 5"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -2.95095887779074, 0, 2.95095887779074"
```

```
print_quant_cts(cts_alpha1, lambda2, c1, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1"
## [1] "параметр mu = 0"
## [1] "квантили = -0.696125824955401, 0, 0.696125824955401"
```

```
print_quant_cts(cts_alpha1, lambda2, c2, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -1.0441887374331, 0, 1.0441887374331"
```

```
print_quant_cts(cts_alpha1, lambda2, c3, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 2"
## [1] "параметр mu = 0"
## [1] "квантили = -1.3922516499108, 0, 1.3922516499108"
```

```
print_quant_cts(cts_alpha1, lambda2, c2, mu1)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = -1"
## [1] "квантили = -2.0441887374331, -1, 0.0441887374331018"
```

```
print_quant_cts(cts_alpha1, lambda2, c2, mu2)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 0"
## [1] "квантили = -1.0441887374331, 0, 1.0441887374331"
```

```
print_quant_cts(cts_alpha1, lambda2, c2, mu3)
```

```
## [1] "параметр alpha = 0.1"
## [1] "параметр C = 1"
## [1] "параметр lambda = 1.5"
## [1] "параметр mu = 1"
## [1] "квантили = -0.0441887374331018, 1, 2.0441887374331"
```

3. $x \sim$ CTS распределение

$$C_1(x) = m + C_+ \Gamma(1-\alpha) \lambda_+^{\alpha-1} - C_- \Gamma(1-\alpha) \lambda_-^{\alpha-1}, \quad n = 1$$

$$C_n(x) = \Gamma(n-\alpha)\left(C_+ \lambda_+^{\alpha-n} + (-1)^n C_- \lambda_-^{\alpha-n}\right), \quad n > 1$$

$$E(x) = C_1(x) = m + C_+ \Gamma(1-\alpha) \lambda_+^{\alpha-1} - C_- \Gamma(1-\alpha) \lambda_-^{\alpha-1}$$

$$V(x) = C_2(x) = \Gamma(2-\alpha)\left(C_+ \lambda_+^{\alpha-2} + (-1)^2 C_- \lambda_-^{\alpha-2}\right)$$

$$S(x) = \frac{C_3(x)}{C_2(x)^{3/2}} = \frac{\Gamma(3-\alpha)\left(C_+ \lambda_+^{\alpha-3} + (-1)^3 C_- \lambda_-^{\alpha-3}\right)}{\left(\Gamma(2-\alpha)\left(C_+ \lambda_+^{\alpha-2} + (-1)^2 C_- \lambda_-^{\alpha-2}\right)\right)^{3/2}}$$

$$K(x) = \frac{C_4(x)}{C_2(x)^2} = \frac{\Gamma(4-\alpha)\left(C_+ \lambda_+^{\alpha-4} + (-1)^4 C_- \lambda_-^{\alpha-4}\right)}{\left(\Gamma(2-\alpha)\left(C_+ \lambda_+^{\alpha-2} + (-1)^2 C_- \lambda_-^{\alpha-2}\right)\right)^2}$$

Задание 2

Описание задания 2

Исследовать зависимость меры возмущений вероятности для рассматриваемых распределений при различных 3-х функциях $g(x)$ и различных параметрах распределения

Для вычисления мер риска будут использоваться следующие формулы:

$$\pi_g^{(1)}(X) = \int_0^\infty g(1 - F_x(t))\ dt$$

$$\pi_g^{(2)}(X) = \int_{-\infty}^0 (g(1 - F_x(t)) - 1)\ dt + \int_0^\infty g(1 - F_x(t))\ dt$$

```
g1 <- function(x) x ^ 2
g2 <- function(x) (x + x ^ 6) / 2
g3 <- function(x)(x + x ^ (6 / 5)) * exp(1 - x) / 2
pi1 <- function(dist_func, g) {
  return(integrate(
    function(t) g(1 - dist_func(t)),
    lower = 0,
    upper = +Inf,
    stop.on.error = FALSE
  )$value)
}
pi2 <- function(dist_func, g) {
  return(integrate(
    function(t) g(1 - dist_func(t)) - 1,
    lower = -Inf,
    upper = 0,
    stop.on.error = FALSE
  )$value + integrate(
    function(t) g(1 - dist_func(t)),
    lower = 0,
    upper = +Inf,
    stop.on.error = FALSE
  )$value)
}
pi_mera <- function(dist_func) {
  return(data.frame(
    pi1.g1 = pi1(dist_func, g1),
    pi1.g2 = pi1(dist_func, g2),
    pi1.g3 = pi1(dist_func, g3),
    pi2.g1 = pi2(dist_func, g1),
    pi2.g2 = pi2(dist_func, g2),
    pi2.g3 = pi2(dist_func, g3)
  ))
}

entire_mera_df <- data.frame()
entire_mera_df <- rbind(entire_mera_df,
  Norm_m3_1 = pi_mera(function(t) pnorm(t, mean = mean1, sd = sd1)),
  Norm_0_1 = pi_mera(function(t) pnorm(t, mean = mean2, sd = sd1)),
  Norm_3_1 = pi_mera(function(t) pnorm(t, mean = mean3, sd = sd1)),
  Norm_0_1 = pi_mera(function(t) pnorm(t, mean = mean2, sd = sd1)),
  Norm_0_0.7 = pi_mera(function(t) pnorm(t, mean = mean2, sd = sd2)),
  Norm_0_0.5 = pi_mera(function(t) pnorm(t, mean = mean2, sd = sd3)),
  Stable_0.5_0_1_0 = pi_mera(function(t) pstable(t, alpha=alpha1, beta=beta2, gamma=g
amma2, delta = delta2)),
  Stable_0.9_0_1_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=g
amma2, delta = delta2)),
  Stable_2_0_1_0 = pi_mera(function(t) pstable(t, alpha=alpha3, beta=beta2, gamma=gam
ma2, delta = delta2)),
  Stable_0.9_0.5_1_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta1, gamma
=gamma2, delta = delta2)),
  Stable_0.9_0_1_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=g
amma2, delta = delta2)),
  Stable_0.9_1_1_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta3, gamma=g
amma2, delta = delta2)),
  Stable_0.9_0_2_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=g
amma1, delta = delta2)),
```

```
  Stable_0.9_0_1_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=g
amma2, delta = delta2)),
  Stable_0.9_0_3_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=g
amma3, delta = delta2)),
  Stable_0.9_0_1_m1 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=
gamma2, delta = delta1)),
  Stable_0.9_0_1_0 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=g
amma2, delta = delta2)),
  Stable_0.9_0_1_1 = pi_mera(function(t) pstable(t, alpha=alpha2, beta=beta2, gamma=g
amma2, delta = delta3))
)
entire_mera_df[,1:3]
```

```
##                           pi1.g1        pi1.g2        pi1.g3
## Norm_m3_1          2.667987e-07  1.910772e-04  0.0006356476
## Norm_0_1           1.168475e-01  2.009043e-01  0.7130115382
## Norm_3_1           2.436574e+00  2.367732e+00  3.5957225802
## Norm_0_11          1.168475e-01  2.009043e-01  0.7130115382
## Norm_0_0.7         8.179324e-02  1.406330e-01  0.4991080822
## Norm_0_0.5         5.842374e-02  1.004521e-01  0.3565057691
## Stable_0.5_0_1_0  -6.707573e+01  4.202309e+03 -3.5478198286
## Stable_0.9_0_1_0   2.471886e-01 -1.541896e+00  9.5259038583
## Stable_2_0_1_0     1.652473e-01  2.841215e-01  1.0083505875
## Stable_0.9_0.5_1_0  4.629453e-01  3.726677e+00 13.4330490749
## Stable_0.9_0_1_01  2.471886e-01 -1.541896e+00  9.5259038583
## Stable_0.9_1_1_0   7.727096e-01  5.462815e+00 19.4240461060
## Stable_0.9_0_2_0   4.943792e-01 -3.083305e+00 19.0519031281
## Stable_0.9_0_1_02  2.471886e-01 -1.541896e+00  9.5259038583
## Stable_0.9_0_3_0   7.415671e-01 -4.625965e+00 27.6561876102
## Stable_0.9_0_1_m1  1.133942e-01 -1.723033e+00  8.9148830274
## Stable_0.9_0_1_03  2.471886e-01 -1.541896e+00  9.5259038583
## Stable_0.9_0_1_1   6.637966e-01 -1.180078e+00 10.3999108830
```

```
entire_mera_df[,4:6]
```

```
##                           pi2.g1        pi2.g2        pi2.g3
## Norm_m3_1         -3.564190e+00 -3.633603e+00 -2.404316e+00
## Norm_0_1          -5.641896e-01 -6.336032e-01  5.956842e-01
## Norm_3_1           2.435810e+00  2.366397e+00  3.595684e+00
## Norm_0_11         -5.641896e-01 -6.336032e-01  5.956842e-01
## Norm_0_0.7        -3.949327e-01 -4.435222e-01  4.169790e-01
## Norm_0_0.5        -2.820948e-01 -3.168016e-01  2.978421e-01
## Stable_0.5_0_1_0  -3.368873e+04 -1.134722e+05 -1.903038e+04
## Stable_0.9_0_1_0   6.671323e+00  1.061067e+01  9.678947e+00
## Stable_2_0_1_0    -7.978846e-01 -8.960502e-01  8.424247e-01
## Stable_0.9_0.5_1_0  3.531908e+00  9.507754e+00  1.348990e+01
## Stable_0.9_0_1_01  6.671323e+00  1.061067e+01  9.678947e+00
## Stable_0.9_1_1_0   3.557460e-01  4.927523e+00  1.936652e+01
## Stable_0.9_0_2_0   1.333921e+01  2.122142e+01  1.935799e+01
## Stable_0.9_0_1_02  6.671323e+00  1.061067e+01  9.678947e+00
## Stable_0.9_0_3_0   2.000881e+01  3.183101e+01  2.811532e+01
## Stable_0.9_0_1_m1  5.669978e+00  9.611597e+00  8.678723e+00
## Stable_0.9_0_1_03  6.671323e+00  1.061067e+01  9.678947e+00
## Stable_0.9_0_1_1   7.669495e+00  1.161015e+01  1.067886e+01
```

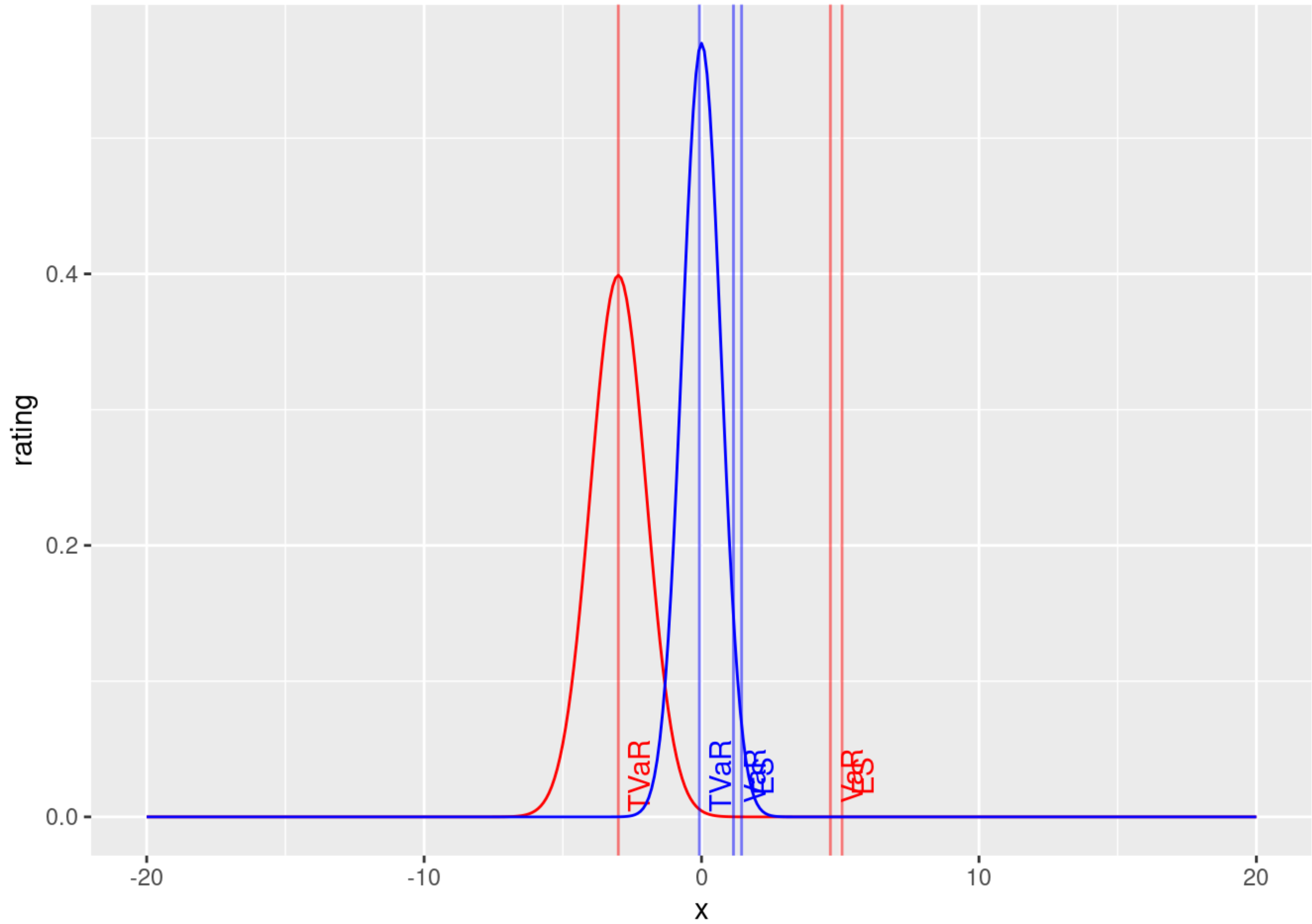
Задание 3

Описание задания 3

Проанализировать на графиках плотностей распределений расположение VaR, TVaR, ES, расположение медианы, моды и мат. ожидания

Графики для нормального распределения

```
get_norm_var_es_tvar <- function(mean, sd){
  num_of_samples <- 1000000
  var <- VaR(qnorm, mean=mean, sd=sd)
  es <- ES(qnorm, mean=mean, sd=sd)
  distr <- rnorm(num_of_samples, mean=mean, sd=sd)
  tvar <- ifelse(min(distr) < 0, mean(distr[distr < var]), mean(distr[distr > var]))
  return(c(var, es, tvar))
}

x <- seq(-20, 20, 0.1)
params1 <- get_norm_var_es_tvar(mean1, sd1)
params2 <- get_norm_var_es_tvar(mean2, sd2)

distr1 <- data.frame(rating = dnorm(x, mean=mean1, sd=sd1), x=x)
distr2 <- data.frame(rating = dnorm(x, mean=mean2, sd=sd2), x=x)


ggplot() +
  geom_line(data=distr1, aes(x=x, y=rating), color = "red") +
  geom_vline(xintercept = params1[1], color="red", alpha=0.5) +
  geom_text(aes(x=params1[1], label="\nVaR", y=0.03, angle=90), colour="red") +
  geom_vline(xintercept = params1[2], color="red", alpha=0.5) +
  geom_text(aes(x=params1[2], label="\nES", y=0.03, angle=90), colour="red") +
  geom_vline(xintercept = params1[3], color="red", alpha=0.5) +
  geom_text(aes(x=params1[3], label="\nTVaR", y=0.03, angle=90), colour="red") +

  geom_line(data=distr2, aes(x=x, y=rating), color = "blue") +
  geom_vline(xintercept = params2[1], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[1], label="\nVaR", y=0.03, angle=90), colour="blue") +
  geom_vline(xintercept = params2[2], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[2], label="\nES", y=0.03, angle=90), colour="blue") +
  geom_vline(xintercept = params2[3], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[3], label="\nTVaR", y=0.03, angle=90), colour="blue")
```


rating
0.4
0.2
0.0
TVaR
TVaR
VaR
ES
VaR
ES
-20
-10
0
10
20
x

Графики для устойчивого распределения

```
get_stable_var_es_tvar <- function(alpha, beta, gamma, delta){
  num_of_samples <- 1000000
  distr <- rstable(num_of_samples, alpha=alpha, beta=beta, gamma=gamma, delta=delta)
  var <- quantile(distr, 0.05)
  es <- ifelse(min(distr) < 0, mean(distr[distr < 0]), mean(distr))
  tvar <- ifelse(min(distr) < 0, mean(distr[distr < var]), mean(distr[distr > var]))
  return(c(var, es, tvar))
}

x <- seq(-30, 30, 0.1)
params1 <- get_stable_var_es_tvar(alpha=alpha1, beta=beta1, gamma=gamma1, delta=delta
1)
params2 <- get_stable_var_es_tvar(alpha=alpha2, beta=beta2, gamma=gamma2, delta=delta
2)

distr1 <- data.frame(rating = dstable(x, alpha=alpha1, beta=beta1, gamma=gamma1, delt
a=delta1), x=x)
distr2 <- data.frame(rating = dstable(x, alpha=alpha2, beta=beta2, gamma=gamma2, delt
a=delta2), x=x)


ggplot() +
  geom_line(data=distr1, aes(x=x, y=rating), color = "red") +
  geom_vline(xintercept = params1[1], color="red", alpha=0.5) +
  geom_text(aes(x=params1[1], label="\nVaR", y=0, angle=90), colour="red") +
  geom_vline(xintercept = params1[2], color="red", alpha=0.5) +
  geom_text(aes(x=params1[2], label="\nES", y=0, angle=90), colour="red") +
  geom_vline(xintercept = params1[3], color="red", alpha=0.5) +
  geom_text(aes(x=params1[3], label="\nTVaR", y=0, angle=90), colour="red") +

  geom_line(data=distr2, aes(x=x, y=rating), color = "blue") +
  geom_vline(xintercept = params2[1], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[1], label="\nVaR", y=0, angle=90), colour="blue") +
  geom_vline(xintercept = params2[2], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[2], label="\nES", y=0, angle=90), colour="blue") +
  geom_vline(xintercept = params2[3], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[3], label="\nTVaR", y=0, angle=90), colour="blue") +
  xlim(-30, 30)
```

```
## Warning: Removed 1 rows containing missing values (geom_vline).
```

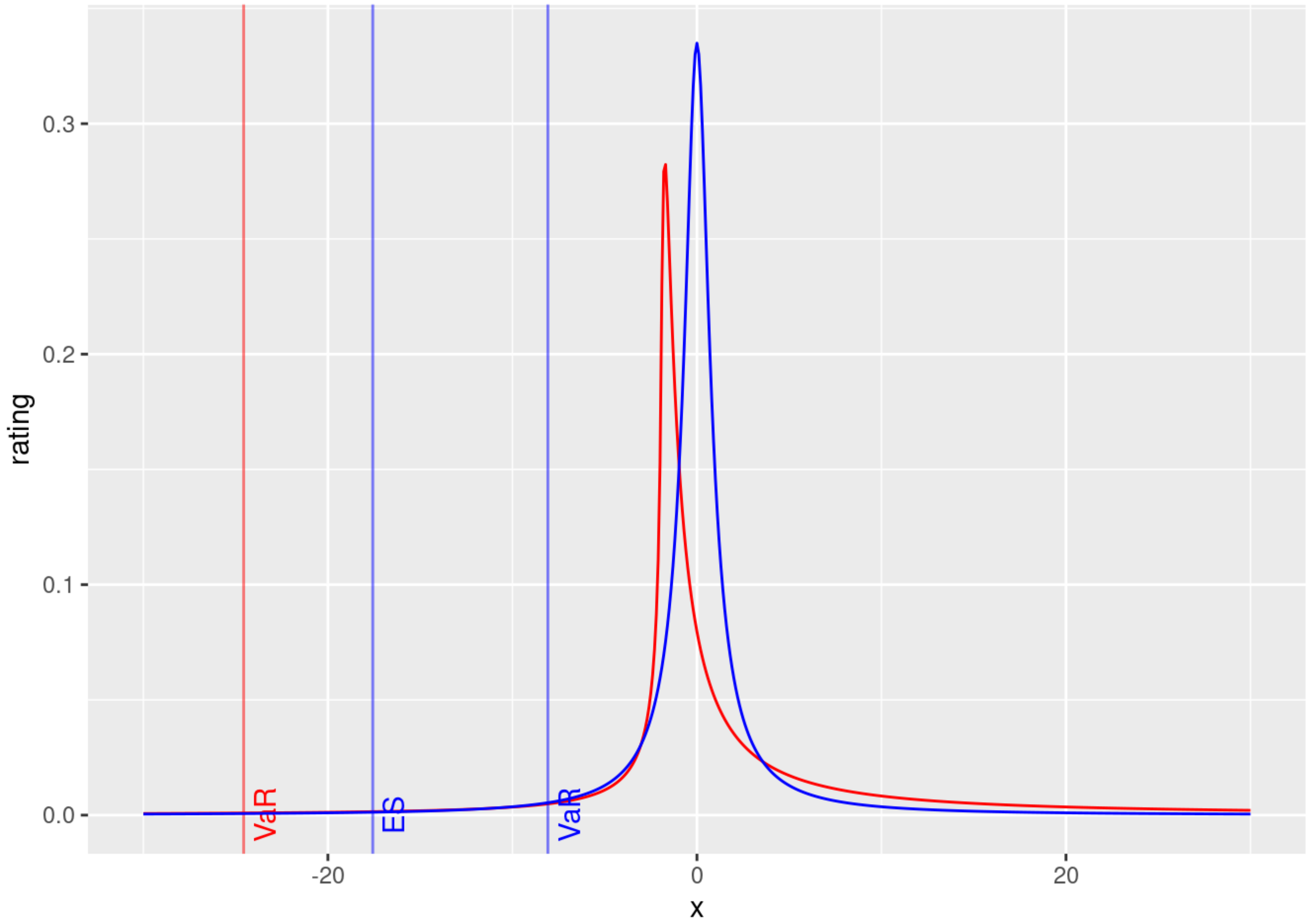
```
## Warning: Removed 1 rows containing missing values (geom_text).
```

```
## Warning: Removed 1 rows containing missing values (geom_vline).
```

```
## Warning: Removed 1 rows containing missing values (geom_text).
```

```
## Warning: Removed 1 rows containing missing values (geom_vline).
```

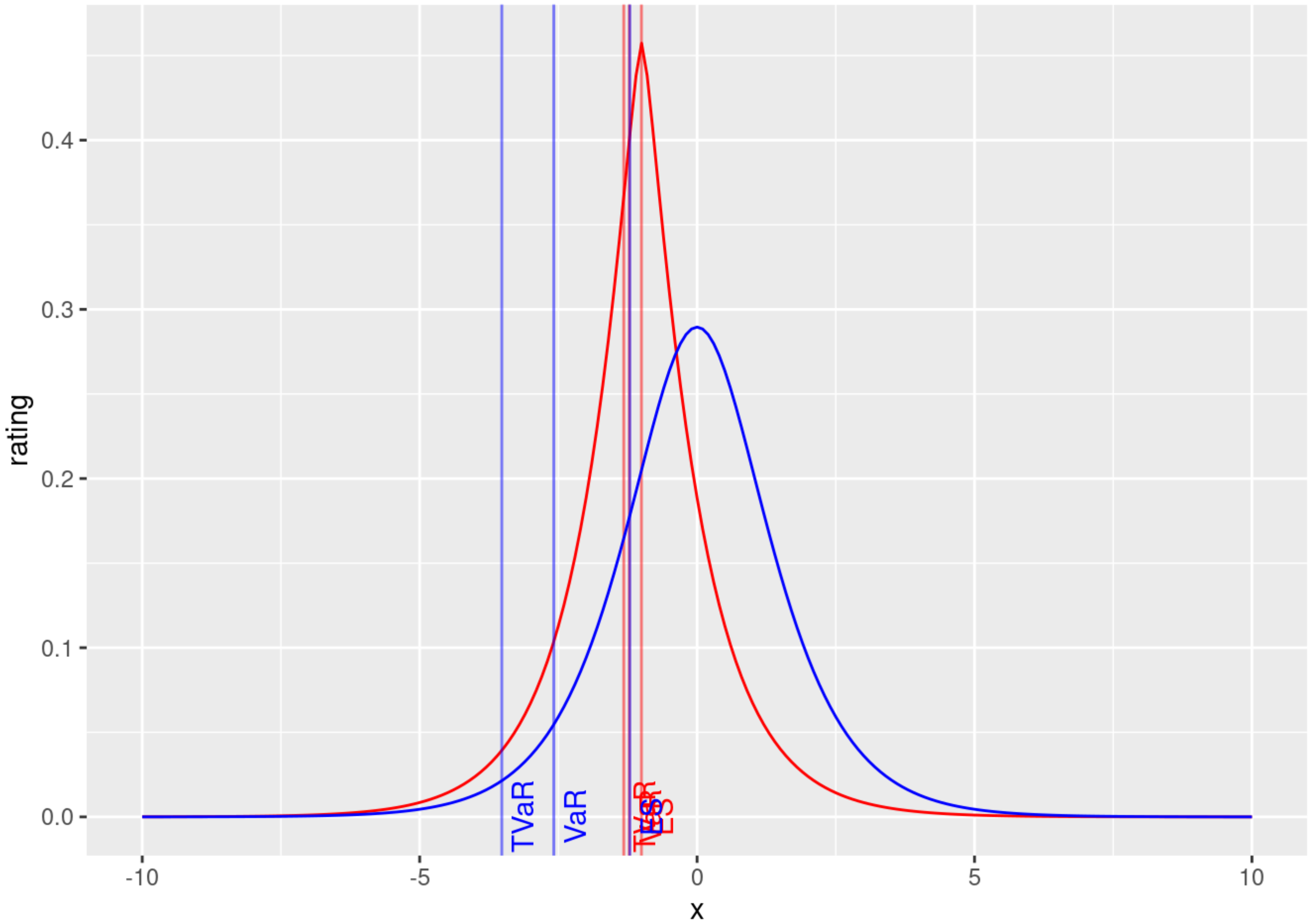
```
## Warning: Removed 1 rows containing missing values (geom_text).
```



Графики для CTS распределения

```
get_cts_var_es_tvar <- function(cts_alpha, c, lambda, mu){
  num_of_samples <- 1000
  distr <- rCTS(num_of_samples, alpha=cts_alpha, c=c, ell=lambda, mu = mu)
  var <- quantile(distr, 0.05)
  es <- ifelse(min(distr) < 0, mean(distr[distr < 0]), mean(distr))
  tvar <- ifelse(min(distr) < 0, mean(distr[distr < var]), mean(distr[distr > var]))
  return(c(var, es, tvar))
}

x <- seq(-30, 30, 0.1)
params1 <- get_cts_var_es_tvar(cts_alpha1, c1, lambda1, mu1)
params2 <- get_cts_var_es_tvar(cts_alpha2, c2, lambda2, mu2)

distr1 <- data.frame(rating = dCTS(x, alpha=cts_alpha1, c=c1, ell=lambda2, mu = mu1),
x=x)
distr2 <- data.frame(rating = dCTS(x, alpha=cts_alpha2, c=c2, ell=lambda2, mu = mu2),
x=x)


ggplot() +
  geom_line(data=distr1, aes(x=x, y=rating), color = "red") +
  geom_vline(xintercept = params1[1], color="red", alpha=0.5) +
  geom_text(aes(x=params1[1], label="\nVaR", y=0, angle=90), colour="red") +
  geom_vline(xintercept = params1[2], color="red", alpha=0.5) +
  geom_text(aes(x=params1[2], label="\nES", y=0, angle=90), colour="red") +
  geom_vline(xintercept = params1[3], color="red", alpha=0.5) +
  geom_text(aes(x=params1[3], label="\nTVaR", y=0, angle=90), colour="red") +

  geom_line(data=distr2, aes(x=x, y=rating), color = "blue") +
  geom_vline(xintercept = params2[1], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[1], label="\nVaR", y=0, angle=90), colour="blue") +
  geom_vline(xintercept = params2[2], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[2], label="\nES", y=0, angle=90), colour="blue") +
  geom_vline(xintercept = params2[3], color="blue", alpha=0.5) +
  geom_text(aes(x=params2[3], label="\nTVaR", y=0, angle=90), colour="blue") +
  xlim(-10, 10)
```

```
## Warning: Removed 400 row(s) containing missing values (geom_path).

## Warning: Removed 400 row(s) containing missing values (geom_path).
```


rating
0.4
0.3
0.2
0.1
0.0
-10
-5
0
5
10
x
TVaR
VaR
TVaR
VaR
ES
ES