

Введение в финансовую математику. Лабораторная работа № 2.

Zhdanov Pavel, C41111

Вариант 12.

Задание 1.

Пусть цена европейского Put опциона с $X = 200\$$ равна $4\$$, а текущая цена акции равна $S = 203\$$. Найдите цену европейского Call опциона с тем же страйком, если $r = 1\%$ и $T - t = 3$. Ответ округлите до целых.

$$C - P = S(t) - X * e^{-r(T-t)}$$

$$C = S(t) - X * e^{-r(T-t)} + P$$

```
P = 4
S = 203
dt = 3
r = 0.01
X = 200

C = S - X * exp(-r*dt) + P
print(round(C))

## [1] 13
```

Задание 2.

При помощи R:

(а) Для европейских опционов построить графики зависимостей

- * цен и payoff-ов от цены базового актива
- * цены Straddle-а от цены базового актива
- * Delta, Gamma, Theta, Vega от цены базового актива
- * Put-Call parity

(b) Пронаблюдать зависимость цены европейского опциона от $T - t$ и от σ , сравнить с Греками.

(с*) Связать график ρ с графиками цены европейских опционов при различных r

Цены европейских put/call опционов задаются формулами:

$$C(S, t) = N(d_1)S - N(d_2)Xe^{-r(T-t)}$$

$$P(S, t) = N(-d_2)Xe^{-r(T-t)} - N(-d_1)S$$

, где

$$d_1 = \frac{1}{\sigma\sqrt{T-t}} * (\ln(\frac{S}{X}) + (r + \frac{\sigma^2}{2}) * (T-t))$$

$$d_2 = d_1 - \sigma * \sqrt{T-t}$$

```
X = 100
sig = 0.05
r = 0.01
dt = 3
spot_price = seq(0, 200, 1)

call_payoff_func <- function(arr){
  n = length(arr)
  ans = rep(0.0, n)
  for( i in c(1:n)){
    if (arr[i] < X){
      ans[i] = 0
    } else {
      ans[i] = arr[i] - X
    }
  }
  return (ans)
}

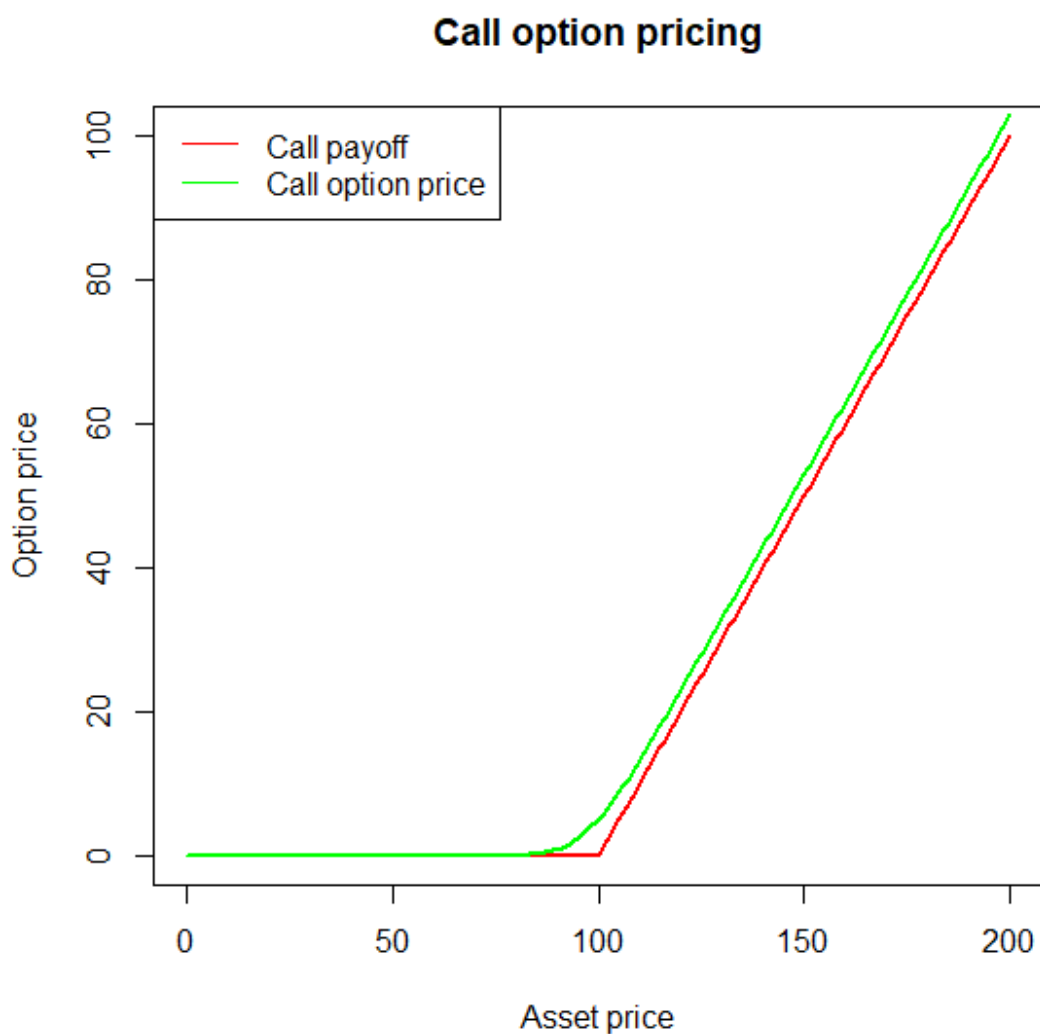
d1 <- function(S, X, sig, r, dt){
  ans = (1/(sig * sqrt(dt))) * (log(S/X) + (r + (sig^2)/2) * dt)
  return (ans)
}

d2 <- function(S, X, sig, r, dt){
  ans = d1(S, X, sig, r, dt) - sig * sqrt(dt)
  return (ans)
}

call_BS_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for (i in c(1:n)){
    ans[i] = pnorm(d1(arr[i], X, sig, r, dt)) * arr[i] - pnorm(d2(arr[i], X, sig
, r, dt)) * X * exp(-r*dt)
  }
  return (ans)
}

plot(spot_price, call_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Call
option pricing')
lines(spot_price, call_BS_func(spot_price, X, sig, r, dt), col=rgb(0.0,1.0,0.0,1
```

```
.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call payoff", "Call option price"), col = c(rgb(1.
0,0.0,0.0,1.0), rgb(0.0,1.0,0.0,1.0)), lty=1:1)
```



```
put_payoff_func <- function(arr){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    if (arr[i] < X){
      ans[i] = X - arr[i]
    } else {
      ans[i] = 0
    }
  }
  return (ans)
}

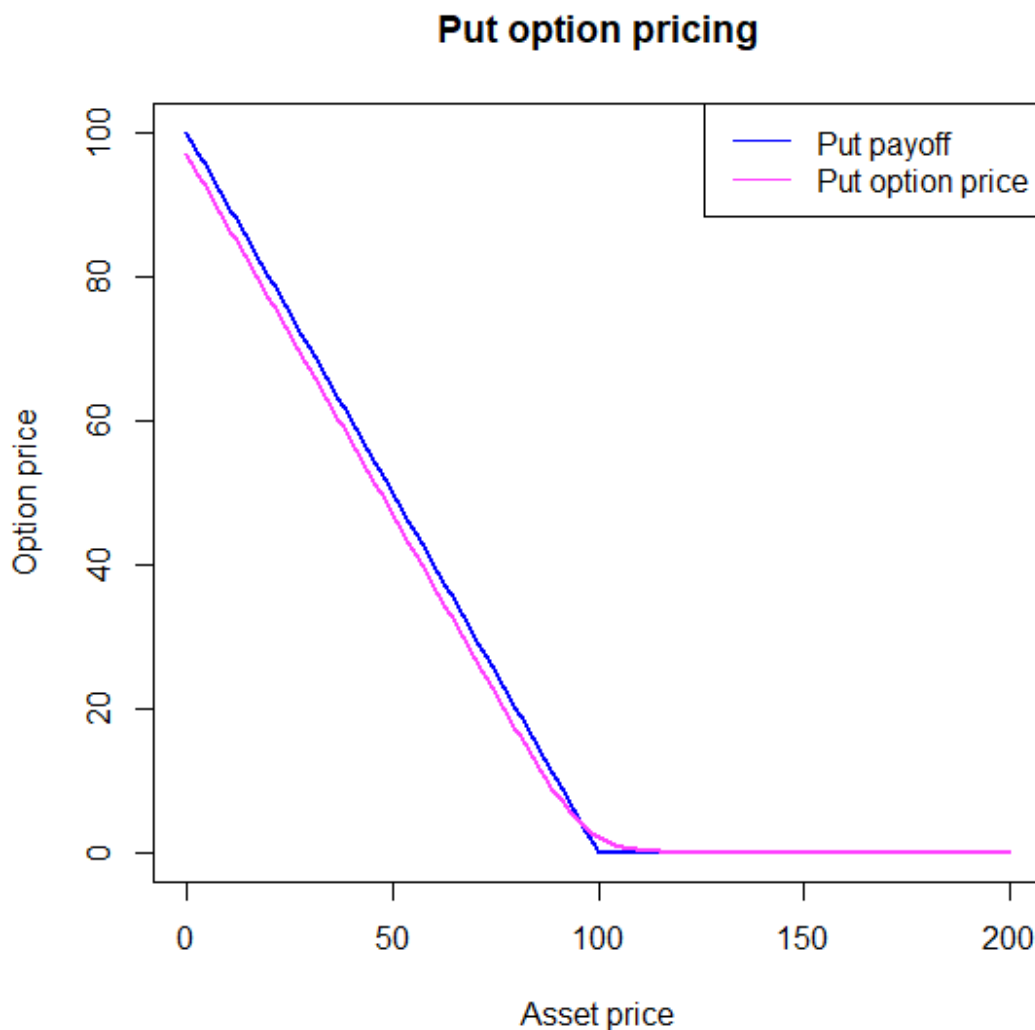
put_BS_func <- function(arr, X, sig, r, dt){
  n = length(arr)
```

```

ans = rep(0.0, n)
for(i in c(1:n)){
  ans[i] = pnorm(-d2(arr[i], X, sig, r, dt)) * X * exp(-r * dt) - pnorm(-d1(arr[i], X, sig, r, dt)) * arr[i]
}
return (ans)
}

plot(spot_price, put_payoff_func(spot_price), col=rgb(0.0,0.0,1.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Put option pricing')
lines(spot_price, put_BS_func(spot_price, X, sig, r, dt), col=rgb(1.0,0.25,1.0,1.0), lwd=2.0, type = 'l')
legend("topright", legend = c("Put payoff", "Put option price"), col = c(rgb(0.0,0.0,1.0,1.0), rgb(1.0,0.25,1.0,1.0)), lty=1:1)

```



```

plot(spot_price, call_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Straddle pricing')

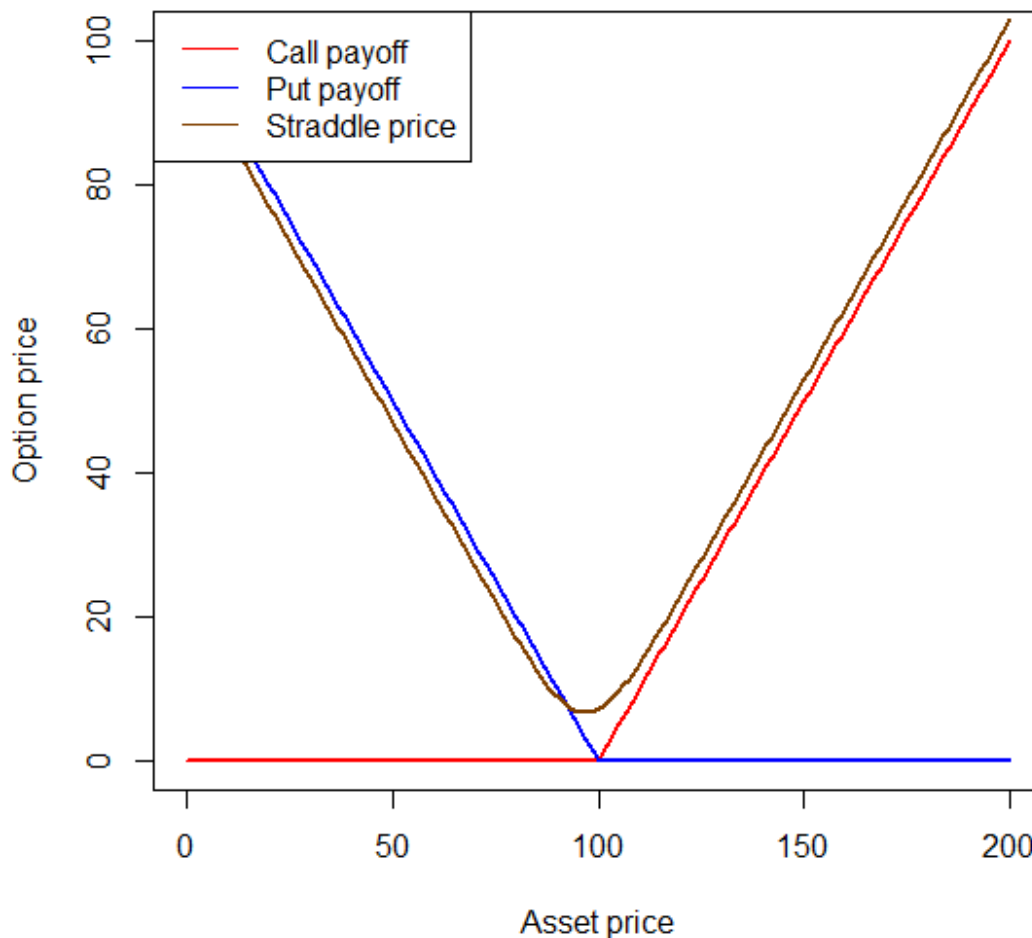
```

```

lines(spot_price, put_payoff_func(spot_price), col=rgb(0.0,0.0,1.0,1.0), lwd=2.0,
      type = 'l')
lines(spot_price, call_BS_func(spot_price, X, sig, r, dt) + put_BS_func(spot_price, X, sig, r, dt), col = rgb(0.5, 0.25, 0.0, 1.0), lwd=2.0, type='l')
legend("topleft", legend = c("Call payoff", "Put payoff", "Straddle price"),
      col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,0.0,1.0,1.0), rgb(0.5, 0.25, 0.0, 1.0)), lty=1:1)

```

Straddle pricing



Формулы для “греков”

1. Call:

$$\Delta = N(d_1)$$

$$\Gamma = \frac{N'(d_1)}{\sigma S \sqrt{(T-t)}}$$

$$\Theta = -\frac{\sigma S N'(d_1)}{2\sqrt{(T-t)}} - r X e^{-r(T-t)} N(d_2)$$

$$Vega = S\sqrt{T-t}N'(d_1)$$

$$Rho = X(T-t)e^{-r(T-t)}N(d_2)$$

2. Put:

$$Delta = N(d_1) - 1$$

$$Gamma = \frac{N'(d_1)}{\sigma S\sqrt{(T-t)}}$$

$$Theta = -\frac{\sigma SN'(-d_1)}{2\sqrt{(T-t)}} + rXe^{-r(T-t)}N(-d_2)$$

$$Vega = S\sqrt{T-t}N'(d_1)$$

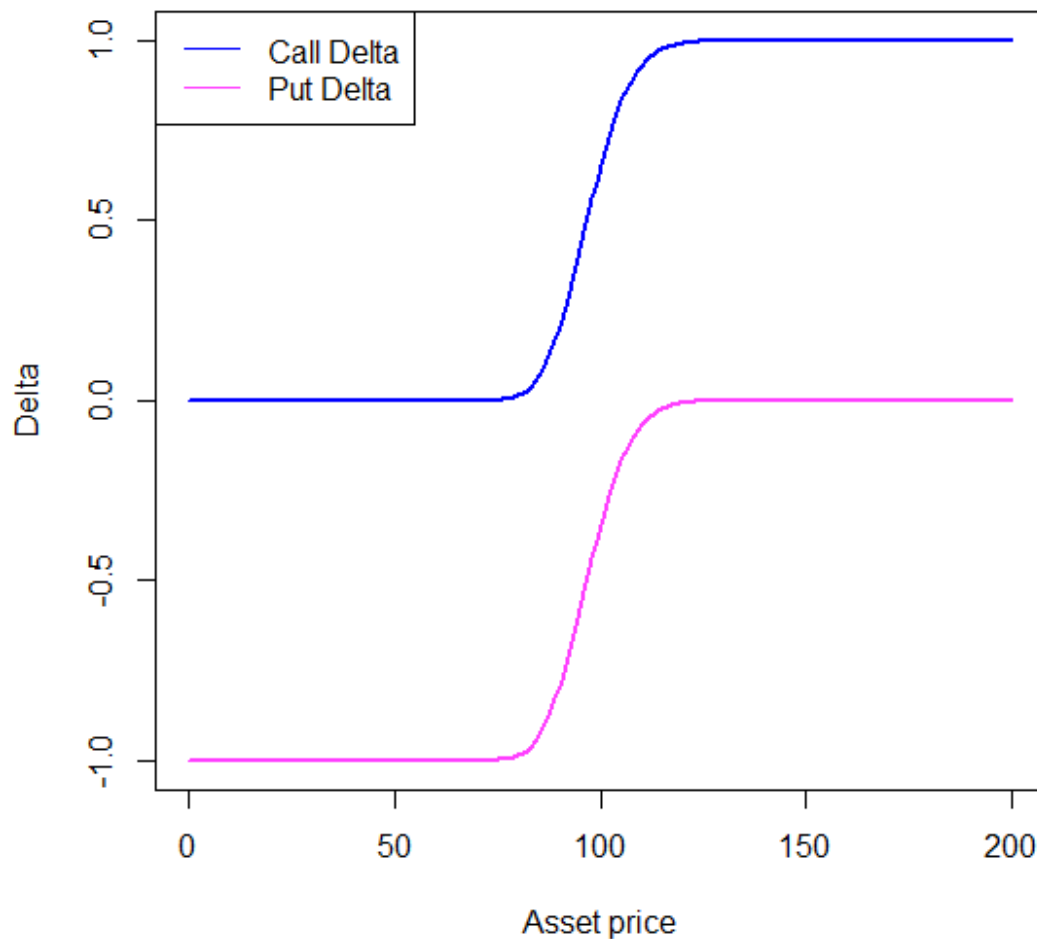
$$Rho = -X(T-t)e^{-r(T-t)}N(-d_2)$$

```
##Delta
call_delta_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = pnorm(d1(arr[i], X, sig, r, dt))
  }
  return (ans)
}

put_delta_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = (pnorm(d1(arr[i], X, sig, r, dt)) - 1)
  }
  return (ans)
}

plot(spot_price, call_delta_func(spot_price, X, sig, r, dt), col=rgb(0.0,0.0,1.0,1.0), ylim=c(-1,1),
      lwd=2.0, type = 'l', ylab = 'Delta', xlab='Asset price', main='Dependency Delta on asset price')
lines(spot_price, put_delta_func(spot_price, X, sig, r, dt), col=rgb(1.0,0.25,1.0,1.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call Delta", "Put Delta"), col = c(rgb(0.0,0.0,1.0,1.0), rgb(1.0,0.25,1.0,1.0)), lty=1:1)
```

Dependency Delta on asset price



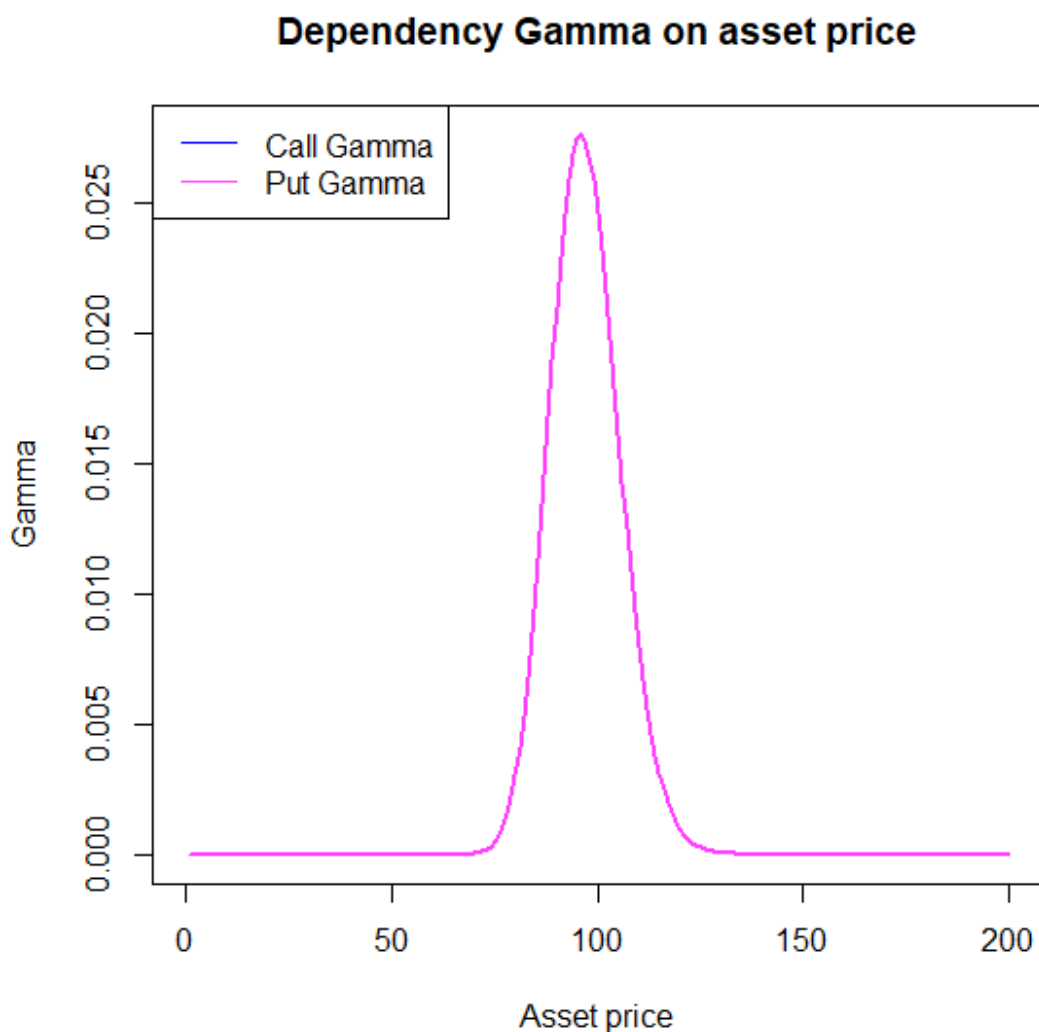
```
##Gamma
call_gamma_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = dnorm(d1(arr[i], X, sig, r, dt))/(sig * arr[i] * dt)
  }
  return (ans)
}

put_gamma_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = dnorm(d1(arr[i], X, sig, r, dt))/(sig * arr[i] * dt)
  }
  return (ans)
}
```

```

plot(spot_price, call_gamma_func(spot_price, X, sig, r, dt), col=rgb(0.0,0.0,1.0
,1.0),
     lwd=2.0, type = 'l', ylab = 'Gamma', xlab='Asset price', main='Dependency G
amma on asset price')
lines(spot_price, put_gamma_func(spot_price, X, sig, r, dt), col=rgb(1.0,0.25,1.
0,1.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call Gamma", "Put Gamma"), col = c(rgb(0.0,0.0,1.0
,1.0), rgb(1.0,0.25,1.0,1.0)), lty=1:1)

```



```

## Theta
call_theta_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = -((sig * arr[i] * dnorm(d1(arr[i], X, sig, r, dt)))/(2*sqrt(dt))) -
r * X * exp(-r * dt) * pnorm(d2(arr[i], X, sig, r, dt))
  }
  return (ans)
}

```



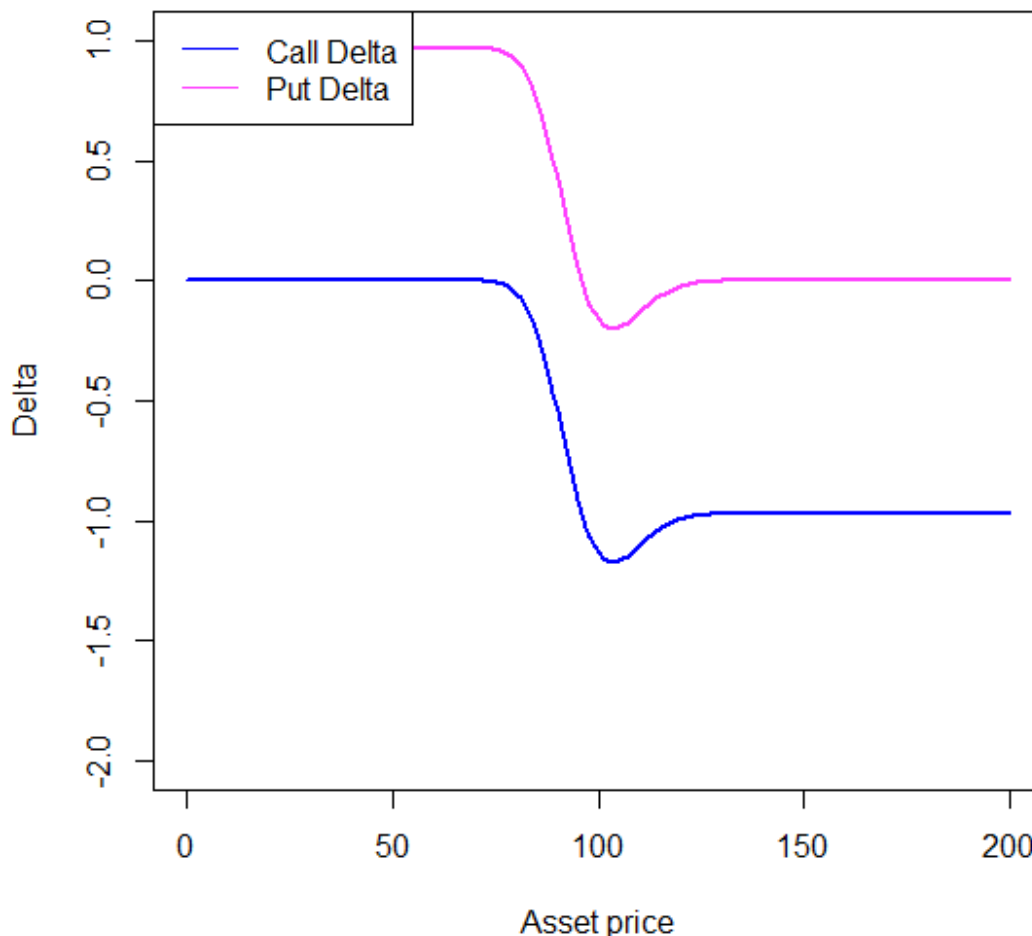
```

put_theta_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = -((sig * arr[i] * dnorm(-d1(arr[i], X, sig, r, dt)))/(2*sqrt(dt)))
+ r * X * exp(-r * dt) * pnorm(-d2(arr[i], X, sig, r, dt))
  }
  return (ans)
}

plot(spot_price, call_theta_func(spot_price, X, sig, r, dt), col=rgb(0.0,0.0,1.0
,1.0), ylim=c(-2,1),
     lwd=2.0, type = 'l', ylab = 'Delta', xlab='Asset price', main='Dependency T
heta on asset price')
lines(spot_price, put_theta_func(spot_price, X, sig, r, dt), col=rgb(1.0,0.25,1.
0,1.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call Delta", "Put Delta"), col = c(rgb(0.0,0.0,1.0
,1.0), rgb(1.0,0.25,1.0,1.0)), lty=1:1)

```

Dependency Theta on asset price



```

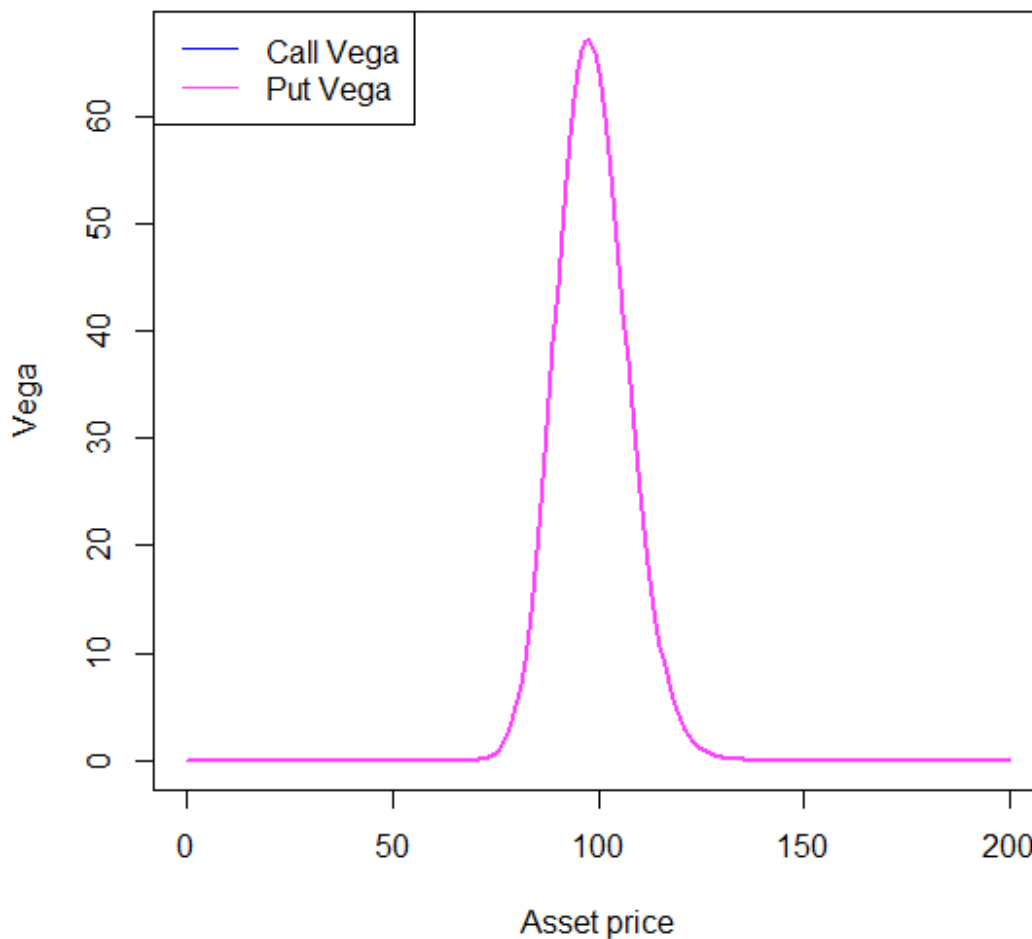
##Vega
call_vega_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = arr[i] * sqrt(dt) * dnorm(d1(arr[i], X, sig, r, dt))
  }
  return (ans)
}

put_vega_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = arr[i] * sqrt(dt) * dnorm(d1(arr[i], X, sig, r, dt))
  }
  return (ans)
}

plot(spot_price, call_vega_func(spot_price, X, sig, r, dt), col=rgb(0.0,0.0,1.0,
1.0),
     lwd=2.0, type = 'l', ylab = 'Vega', xlab='Asset price', main='Dependency Ve
ga on asset price')
lines(spot_price, put_vega_func(spot_price, X, sig, r, dt), col=rgb(1.0,0.25,1.0
,1.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call Vega", "Put Vega"), col = c(rgb(0.0,0.0,1.0,1
.0), rgb(1.0,0.25,1.0,1.0)), lty=1:1)

```

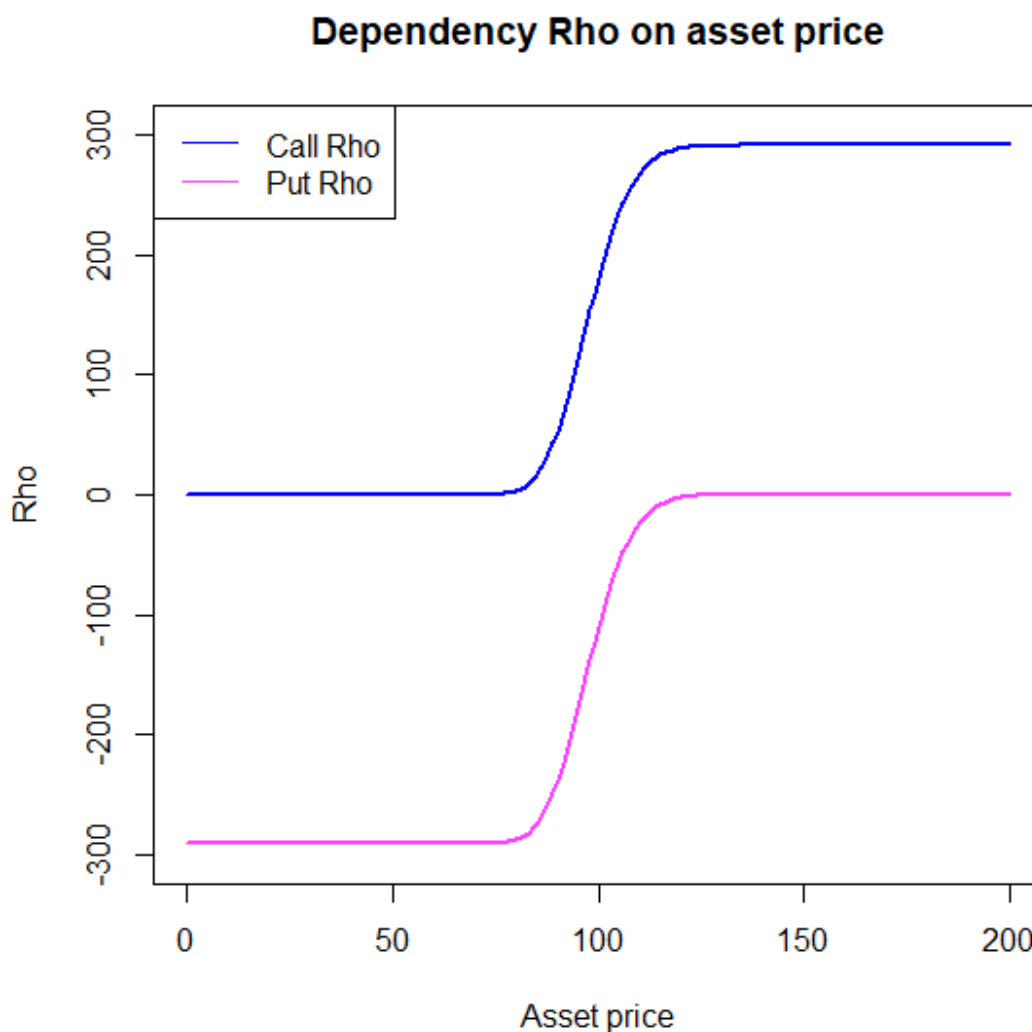
Dependency Vega on asset price



```
##Rho
call_rho_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = X * dt * exp(-r * dt)*pnorm(d2(arr[i], X, sig, r, dt))
  }
  return (ans)
}

put_rho_func <- function(arr, X, sig, r, dt){
  n = length(arr)
  ans = rep(0.0, n)
  for(i in c(1:n)){
    ans[i] = -X * dt * exp(-r * dt)*pnorm(-d2(arr[i], X, sig, r, dt))
  }
  return (ans)
}
```

```
plot(spot_price, call_rho_func(spot_price, X, sig, r, dt), col=rgb(0.0,0.0,1.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Rho', xlab='Asset price', main='Dependency Rho
on asset price', ylim = c(-300, 300))
lines(spot_price, put_rho_func(spot_price, X, sig, r, dt), col=rgb(1.0,0.25,1.0,
1.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call Rho", "Put Rho"), col = c(rgb(0.0,0.0,1.0,1.0
), rgb(1.0,0.25,1.0,1.0)), lty=1:1)
```



Выводы по “грекам”:

1. Delta: измеряет скорость изменения теоретической стоимости опциона по отношению к изменениям цены базового актива. Является первой производной теоретической стоимости опциона по отношению к цене базового инструмента. Разница между call и put равна 1, что является следствием put-call parity: long call + short put повторяет форвард, дельта которого равна 1.
2. Gamma: представляет собой вторую производную цены опциона по базовому активу. Измеряет скорость изменения дельты относительно изменений базовой цены. График одинаков для опционов put и call, потому что для них delta растет одинаково.

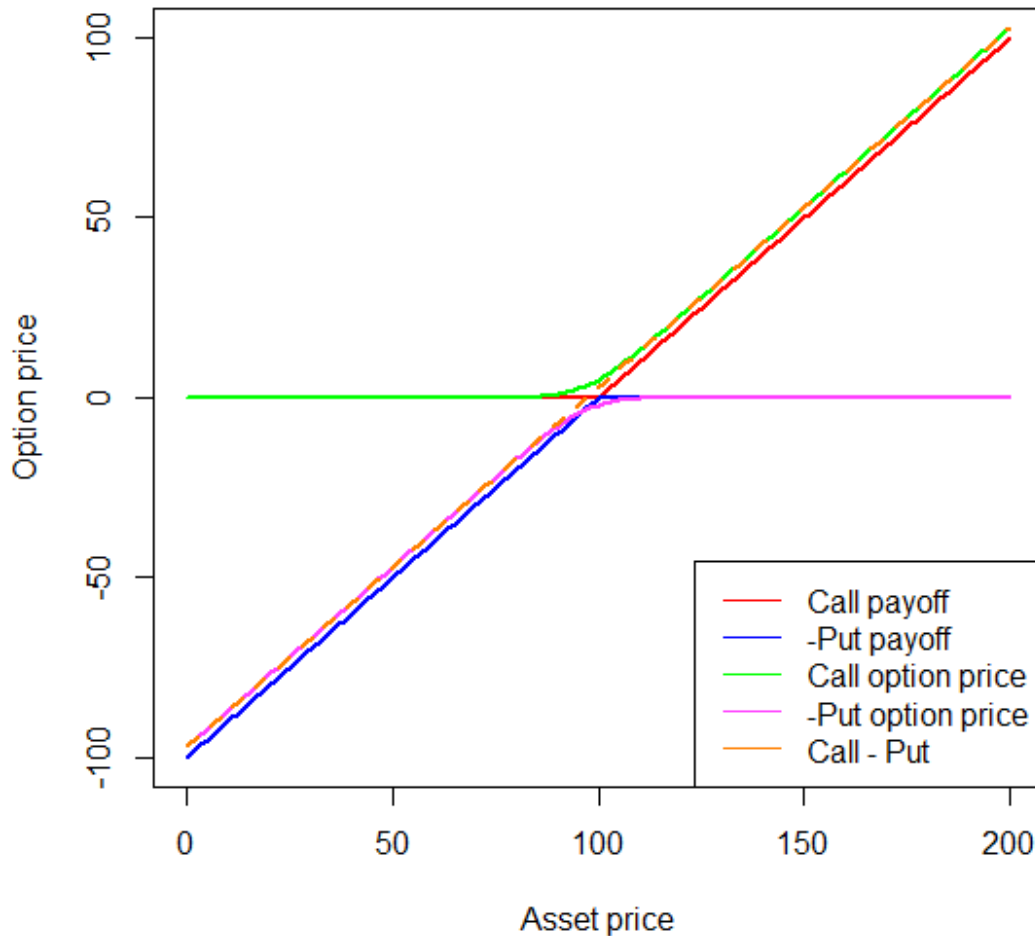
3. Theta: представляет собой производную цены опциона по времени. По мере приближения времени к дате экспирации цена call опциона уменьшается, а put опциона увеличивается.

4. Vega: представляет собой производную цены опциона по отношению к волатильности базового актива. График одинаков для опционов put и call, потому что цена опциона изменяется с одной и той же скоростью при изменении волатильности.

5. Rho: представляет собой производную цены опциона по безрисковой процентной ставке. С увеличением процентной ставки цена опциона call растет, а цена опциона put падает. Это можно объяснить тем, что рост ставки приводит к увеличению ожидаемого дохода от акции, что ведет к увеличению стоимости опциона call и к уменьшению стоимости опциона put.

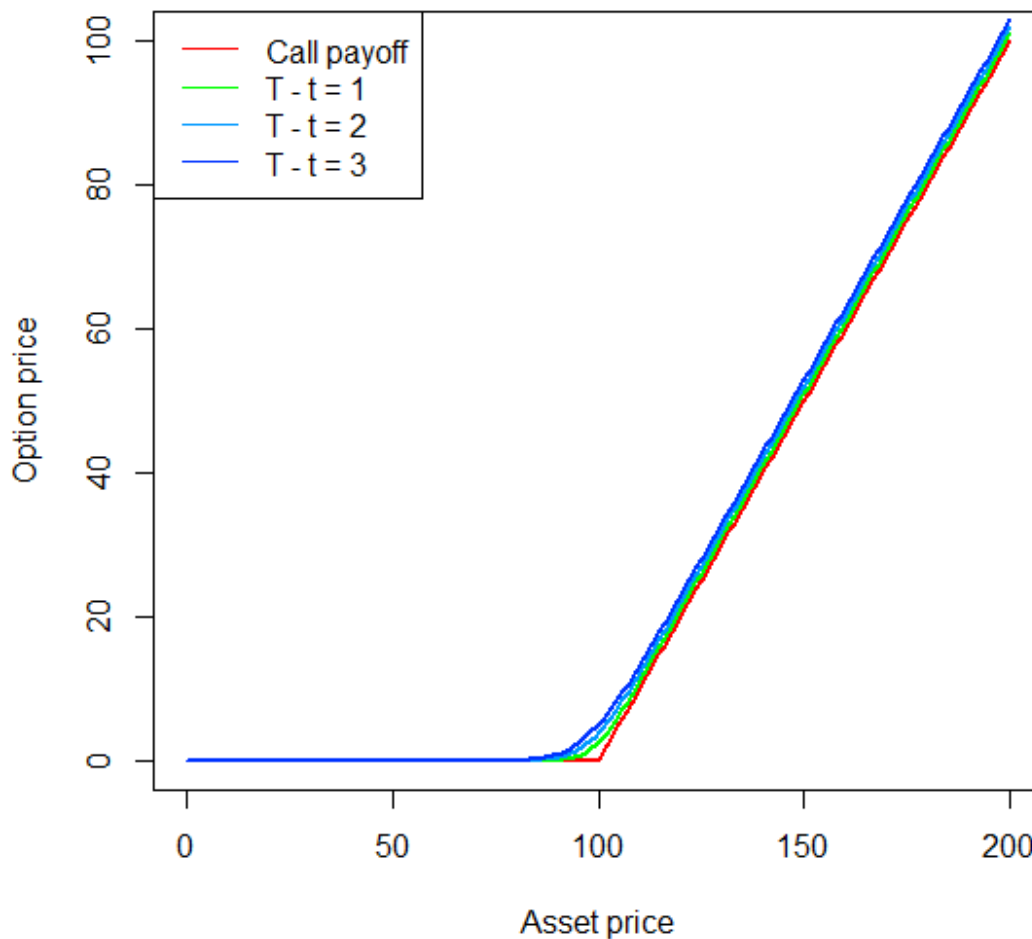
```
# Put-call parity
plot(spot_price, call_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0), ylim =
c(-100, 100),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Put-c
all parity')
lines(spot_price, -put_payoff_func(spot_price), col=rgb(0.0,0.0,1.0,1.0), lwd=2.
0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, sig, r, dt), col=rgb(0.0,1.0,0.0,1
.0), lwd=2.0, type = 'l')
lines(spot_price, -put_BS_func(spot_price, X, sig, r, dt), col=rgb(1.0,0.25,1.0,
1.0), lwd=2.0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, sig, r, dt) - put_BS_func(spot_pri
ce, X, sig, r, dt), col = rgb(1.0, 0.5, 0.0, 1.0), lwd=2.0, lty = 2, type='l')
legend("bottomright", legend = c("Call payoff", "-Put payoff", "Call option pric
e", "-Put option price", "Call - Put"),
     col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,0.0,1.0,1.0), rgb(0.0,1.0,0.0,1.0),
rgb(1.0,0.25,1.0,1.0), rgb(1.0, 0.5, 0.0, 1.0)), lty=1:1)
```

Put-call parity



```
plot(spot_price, call_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Depen
dency of call option prices on (T - t)')
lines(spot_price, call_BS_func(spot_price, X, sig, r, 1), col=rgb(0.0,1.0,0.0,1.
0), lwd=2.0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, sig, r, 2), col=rgb(0.0,0.6,1.0,1.
0), lwd=2.0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, sig, r, 3), col=rgb(0.0,0.2,1.0,1.
0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call payoff", "T - t = 1", "T - t = 2", "T - t = 3
"), col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,1.0,0.0,1.0), rgb(0.0,0.6,1.0,1.0), rg
b(0.0,0.2,1.0,1.0)), lty=1:1)
```

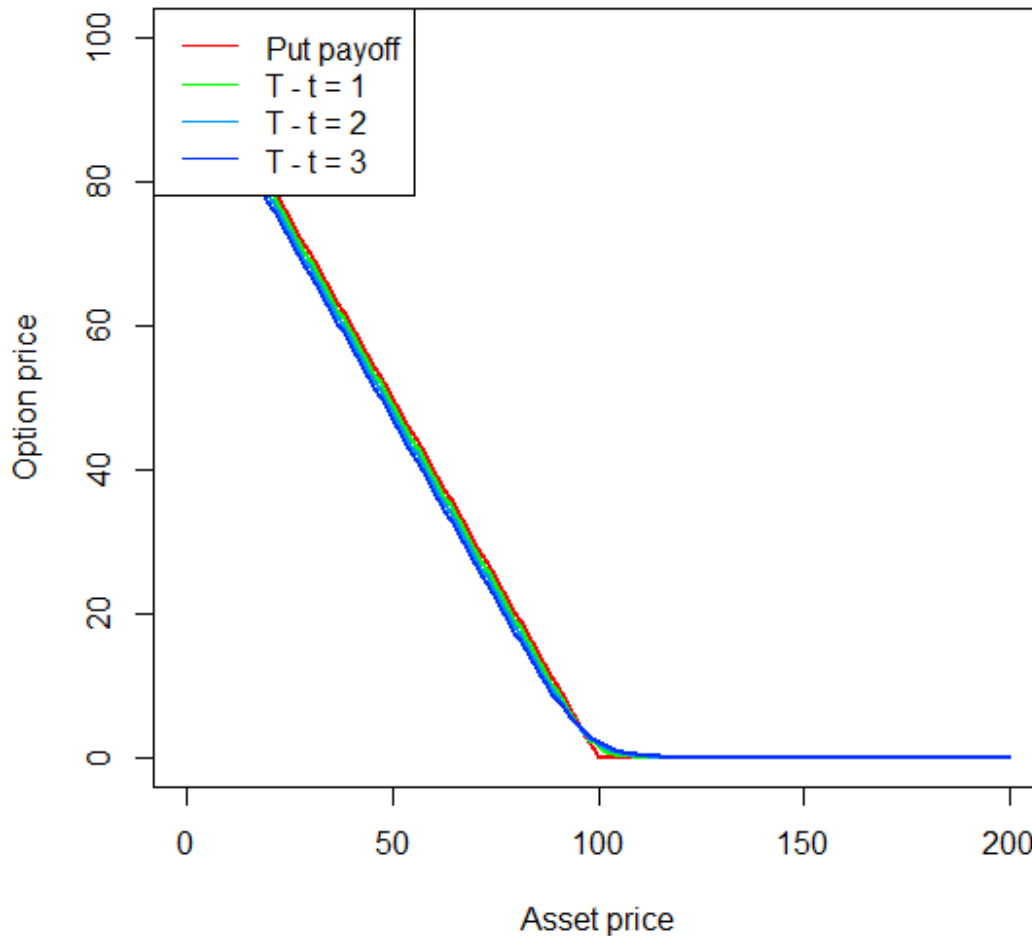
Dependency of call option prices on (T - t)



```
plot(spot_price, put_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Depen
dependency of put option prices on (T - t)')
lines(spot_price, put_BS_func(spot_price, X, sig, r, 1), col=rgb(0.0,1.0,0.0,1.0
), lwd=2.0, type = 'l')
lines(spot_price, put_BS_func(spot_price, X, sig, r, 2), col=rgb(0.0,0.6,1.0,1.0
), lwd=2.0, type = 'l')
lines(spot_price, put_BS_func(spot_price, X, sig, r, 3), col=rgb(0.0,0.2,1.0,1.0
), lwd=2.0, type = 'l')

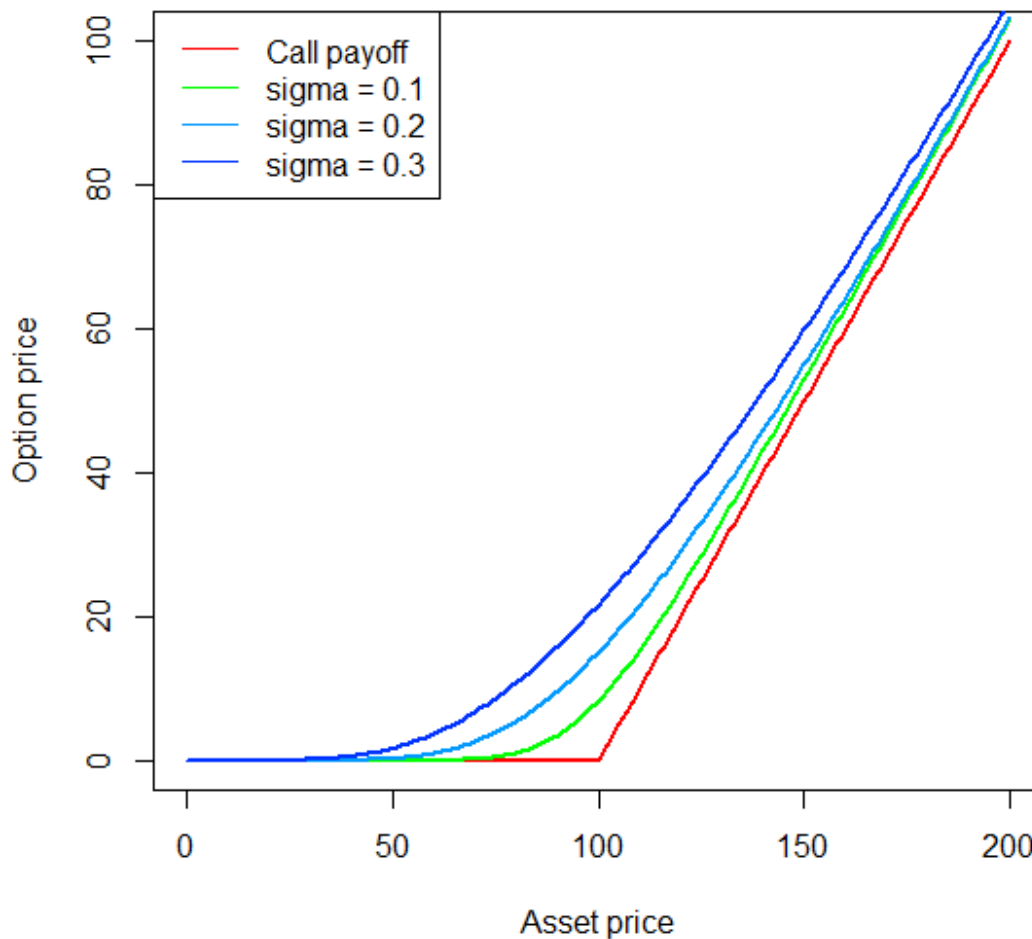
legend("topleft", legend = c("Put payoff", "T - t = 1", "T - t = 2", "T - t = 3"
), col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,1.0,0.0,1.0), rgb(0.0,0.6,1.0,1.0), rgb
(0.0,0.2,1.0,1.0)), lty=1:1)
```

Dependency of put option prices on (T - t)



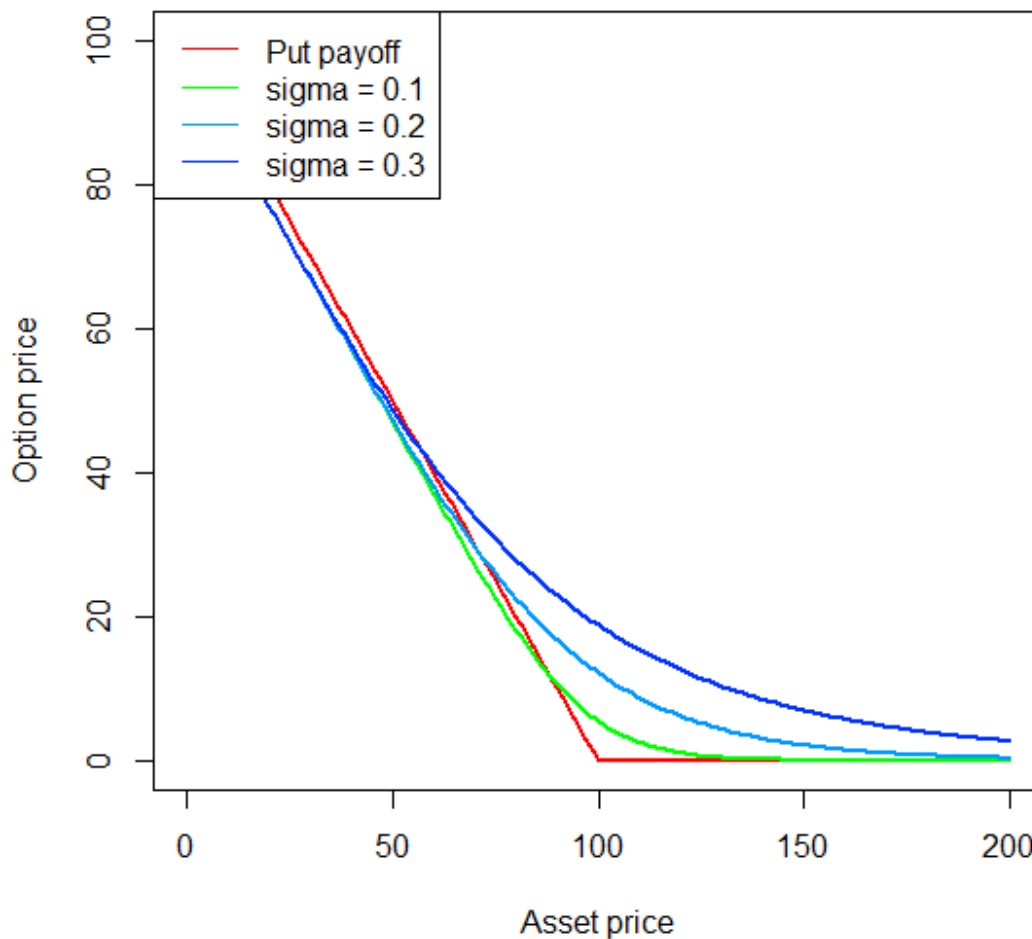
```
plot(spot_price, call_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Depen
dency of call option prices on sigma')
lines(spot_price, call_BS_func(spot_price, X, 0.1, r, dt), col=rgb(0.0,1.0,0.0,1
.0), lwd=2.0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, 0.2, r, dt), col=rgb(0.0,0.6,1.0,1
.0), lwd=2.0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, 0.3, r, dt), col=rgb(0.0,0.2,1.0,1
.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call payoff", "sigma = 0.1", "sigma = 0.2", "sigma
= 0.3"), col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,1.0,0.0,1.0), rgb(0.0,0.6,1.0,1.0
), rgb(0.0,0.2,1.0,1.0)), lty=1:1)
```


Dependency of call option prices on sigma



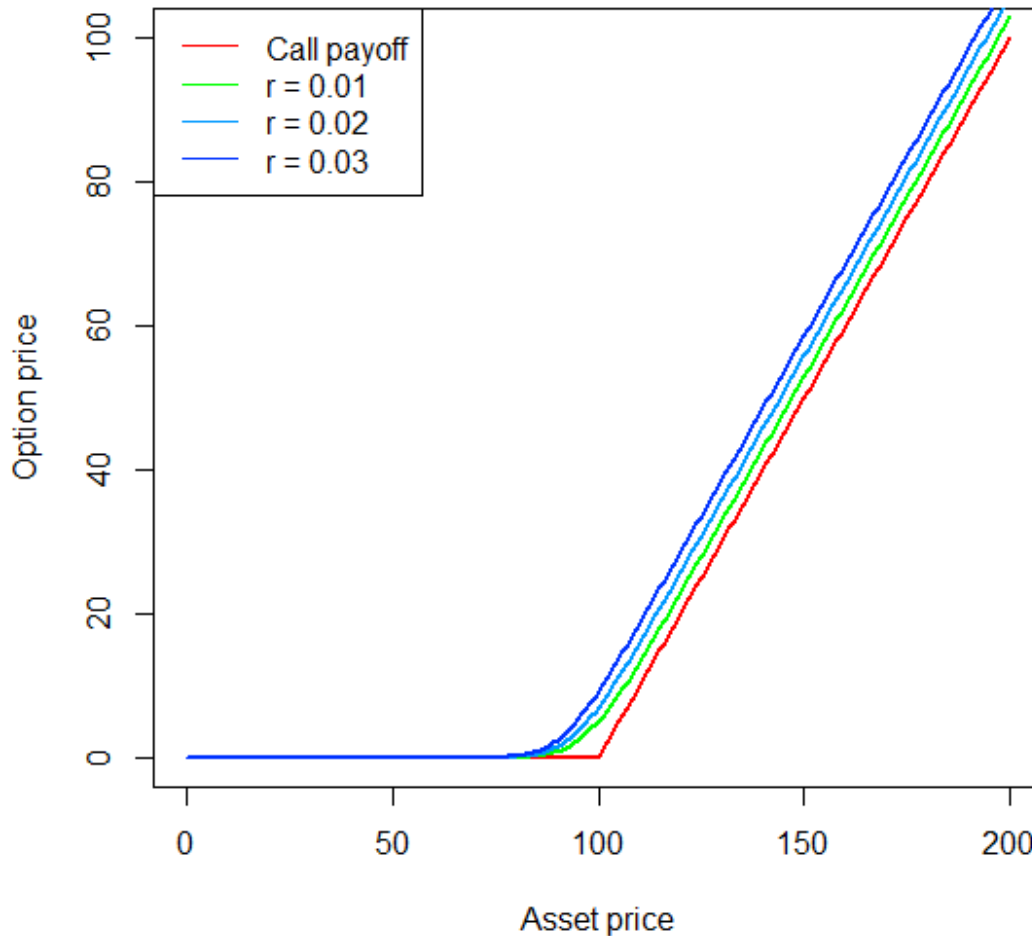
```
plot(spot_price, put_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Depen
dency of put option prices on sigma')
lines(spot_price, put_BS_func(spot_price, X, 0.1, r, dt), col=rgb(0.0,1.0,0.0,1.
0), lwd=2.0, type = 'l')
lines(spot_price, put_BS_func(spot_price, X, 0.2, r, dt), col=rgb(0.0,0.6,1.0,1.
0), lwd=2.0, type = 'l')
lines(spot_price, put_BS_func(spot_price, X, 0.3, r, dt), col=rgb(0.0,0.2,1.0,1.
0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Put payoff", "sigma = 0.1", "sigma = 0.2", "sigma
= 0.3"), col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,1.0,0.0,1.0), rgb(0.0,0.6,1.0,1.0
), rgb(0.0,0.2,1.0,1.0)), lty=1:1)
```

Dependency of put option prices on sigma



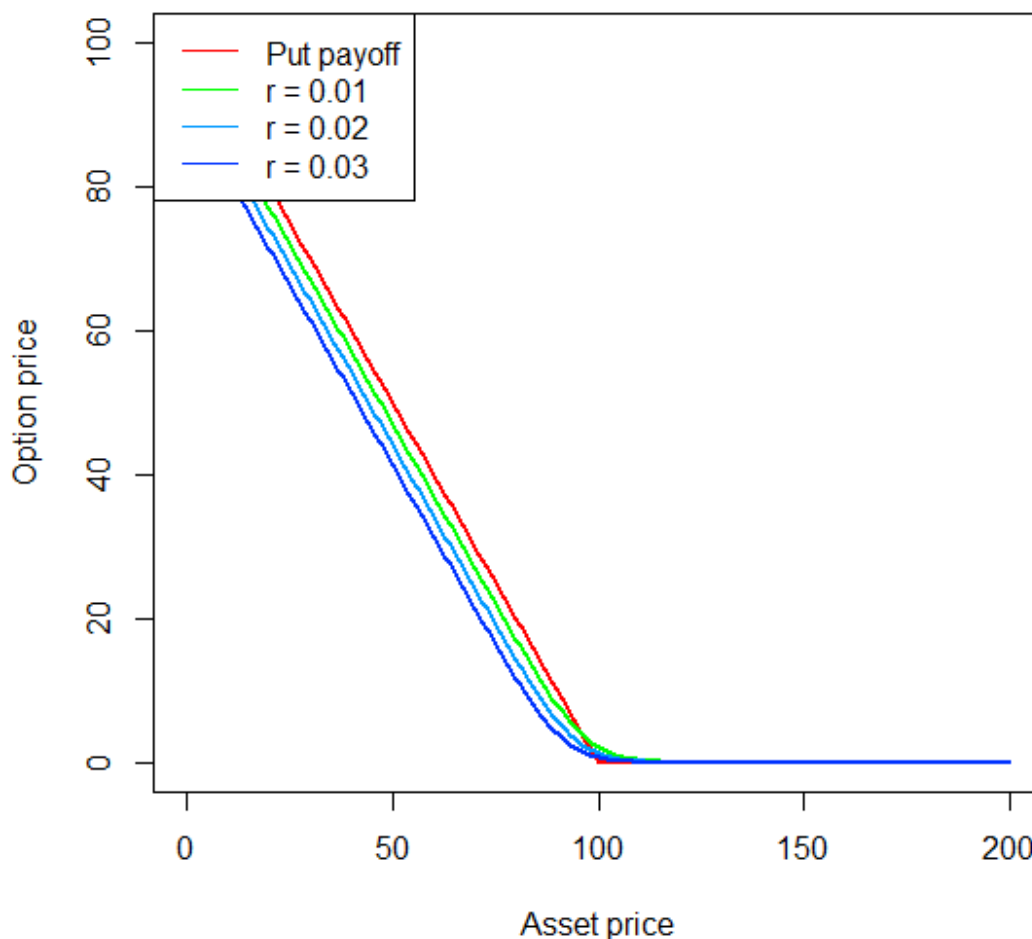
```
plot(spot_price, call_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Depen
dency of call option prices on r')
lines(spot_price, call_BS_func(spot_price, X, sig, 0.01, dt), col=rgb(0.0,1.0,0.
0,1.0), lwd=2.0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, sig, 0.02, dt), col=rgb(0.0,0.6,1.
0,1.0), lwd=2.0, type = 'l')
lines(spot_price, call_BS_func(spot_price, X, sig, 0.03, dt), col=rgb(0.0,0.2,1.
0,1.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Call payoff", "r = 0.01", "r = 0.02", "r = 0.03"),
col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,1.0,0.0,1.0), rgb(0.0,0.6,1.0,1.0), rgb(0.
0,0.2,1.0,1.0)), lty=1:1)
```

Dependency of call option prices on r



```
plot(spot_price, put_payoff_func(spot_price), col=rgb(1.0,0.0,0.0,1.0),
     lwd=2.0, type = 'l', ylab = 'Option price', xlab='Asset price', main='Depen
dency of put option prices on r')
lines(spot_price, put_BS_func(spot_price, X, sig, 0.01, dt), col=rgb(0.0,1.0,0.0
,1.0), lwd=2.0, type = 'l')
lines(spot_price, put_BS_func(spot_price, X, sig, 0.02, dt), col=rgb(0.0,0.6,1.0
,1.0), lwd=2.0, type = 'l')
lines(spot_price, put_BS_func(spot_price, X, sig, 0.03, dt), col=rgb(0.0,0.2,1.0
,1.0), lwd=2.0, type = 'l')
legend("topleft", legend = c("Put payoff", "r = 0.01", "r = 0.02", "r = 0.03"),
col = c(rgb(1.0,0.0,0.0,1.0), rgb(0.0,1.0,0.0,1.0), rgb(0.0,0.6,1.0,1.0), rgb(0.
0,0.2,1.0,1.0)), lty=1:1)
```

Dependency of put option prices on r



Выводы по зависимости цены опциона от параметров уравнения:

1. Чем больше $(T - t)$ тем меньше цена опциона call и больше цена опциона put. Те же выводы были получены для опционов по графикам для Theta.
2. Чем больше сигма, тем больше цена опционов call и put. Те же выводы были получены для опционов по графикам для Vega.
3. Чем больше r , тем больше цена опциона call и тем меньше цена опциона put. Те же выводы были получены для опционов по графикам для Rho.