

PREZENTACJA PROJEKTU

Grzegorz Kupeć, Natalia Słomka, Remigiusz Sroka, Adam Wawrzyńczyk

CO BĘDZIEMY ANALIZOWAĆ?

- Naszym celem będzie analiza danych giełdowych
- W naszych danych mamy datę, wartość na otwarciu, wartość maksymalną, wartość najniższą, wartość na zamknięciu, wolumen

PRZYKŁADOWY FRAGMENT TABELI Z DANYMI

Data	Otwarcie	Najwyższy	Najniższy	Zamknięcie	Wolumen
2007-01-02	3288.38	3352.81	3288.38	3347.75	26027084
2007-01-03	3337.32	3358.26	3325.73	3342.86	17815784
2007-01-04	3286.27	3291.4	3261.26	3276.78	19233714
2007-01-05	3260.11	3269.99	3191.57	3202.15	35440708
2007-01-08	3172.24	3214.43	3144.5	3197.61	20263560
2007-01-09	3219.45	3239.2	3190.33	3198.45	21695212
2007-01-10	3177.78	3198.2	3139.19	3148.68	18925580
2007-01-11	3201.28	3241.39	3187.84	3240.3	28247384
2007-01-12	3240.51	3267.11	3229.26	3262.98	21450942

CO CHCEMY UZYSKAĆ Z DANYCH?

- Będziemy chcieli zobaczyć, jak wyglądała trajektoria WIG20 w przeszłości oraz wykonać symulację przyszłych trajektorii
- Policzymy również kwantyle symulacji przyszłych trajektorii WIG20
- Uzyskamy także tzw. giełdowe wskaźniki greckie

FUNKCJE ANALIZUJĄCE GIEŁDĘ - OMÓWIENIE

```
library(readr)
library(ggplot2)

### Pobieranie danych o giełdzie z założeniem, że ostatni wiersz to ostatni dzień, który będziemy symulować.
stock_data <- read_csv("wig20_dane_historyczne.csv")

### Funkcja zwracająca zwroty oraz dryf i zmienność wyliczoną na podstawie zwrotów

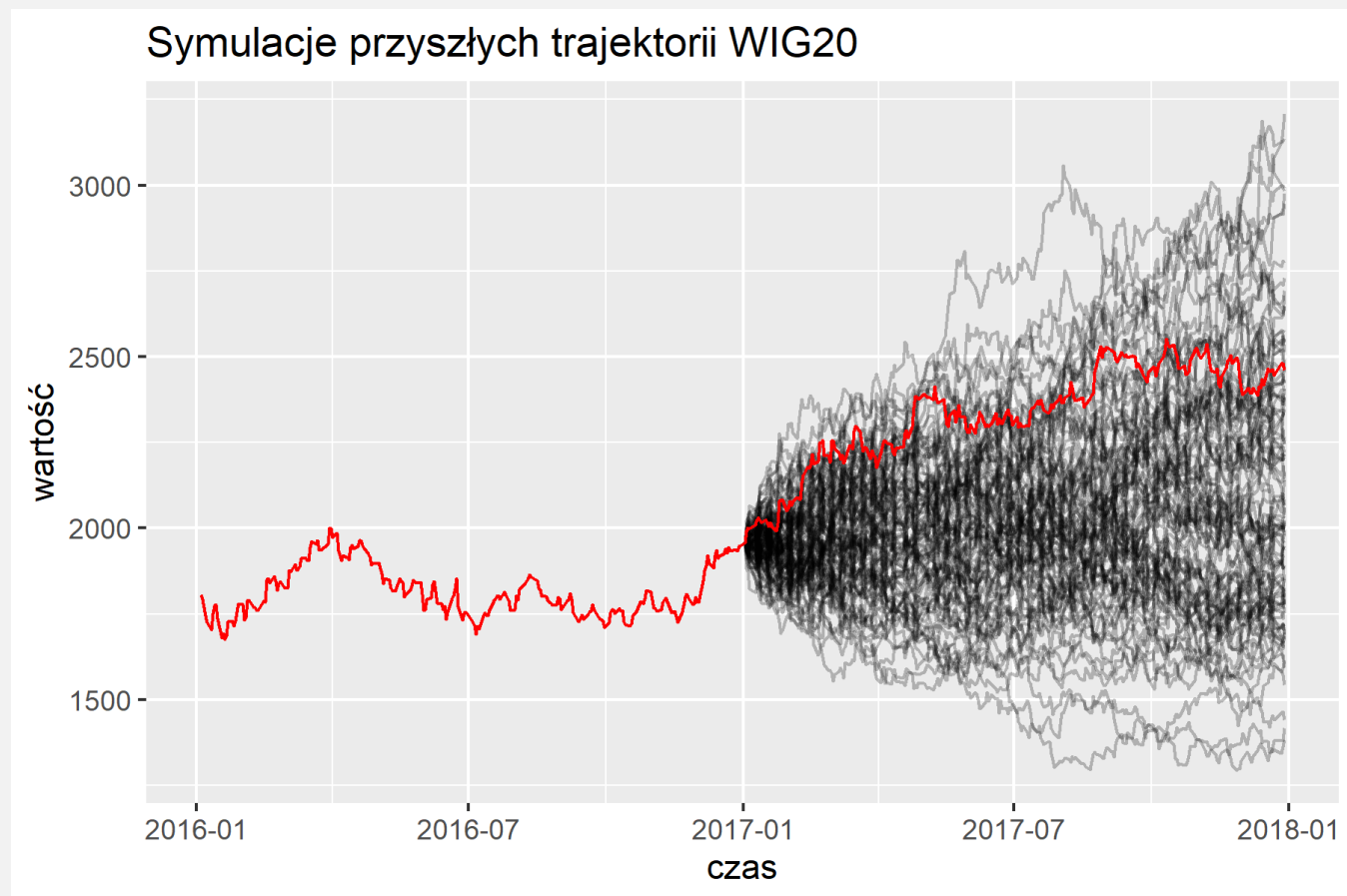
zwroty <- function(start_date, end_date, T_)
{
  nr_row_start_date <- which(stock_data$Data == start_date)
  nr_row_end_date <- which(stock_data$Data == end_date)
  dt <- T_/length(stock_data$Data[nr_row_start_date:nr_row_end_date]) #delta t
  zwrot <- rep(0, times = (length(stock_data$Data[nr_row_start_date:nr_row_end_date]) - 1))
  for (i in 1:length(zwrot))
  {
    zwrot[i] = (stock_data$Zamkniecie[nr_row_start_date + i] - stock_data$Zamkniecie[nr_row_start_date + i - 1])/stock_data$Zamkniecie[nr_row_start_date + i - 1]
  }
  dryf <- mean(zwrot)/dt
  zmienosc <- sqrt(1/dt)*sd(zwrot)
  return(list(zwroty, dryf, zmienosc))
}
```

FUNKCJE ANALIZUJĄCE GIEŁDĘ - DALSZĄ CZĘŚĆ

```
24 dryf <- zwroty("2016-01-04", "2016-12-30", 1)[[2]]
25 zmienosc <- zwroty("2016-01-04", "2016-12-30", 1)[[3]]
26
27 #macierz symulacji akcji
28 macierzAkcjaSim <- function(s_0, mu, sigma, n_days, n_sim, T_){
29   mean = mu*(T_/n_days) - (sigma^2*(T_/n_days))/2
30   sd = sigma*sqrt(T_/n_days)
31   baz = rnorm(n_days*n_sim, mean, sd)
32   tr = matrix(exp(baz),n_days, n_sim)
33   tr = rbind(rep(1, times = n_sim), tr)
34   tr = s_0 * apply(tr,2,cumprod)
35   t(tr)
36 }
37
38 ### wartość indeksu od którego zaczynamy symulować trajektorie
39 stock_value <- as.numeric(stock_data[which(stock_data$Data == "2017-01-02"), "Zamknięcie"])
40
41 ### Symulacje
42 n_sim <- 100 #ilość symulacji
43 simulations <- macierzAkcjaSim(stock_value, dryf, zmienosc, 249, n_sim, 1)
44
```

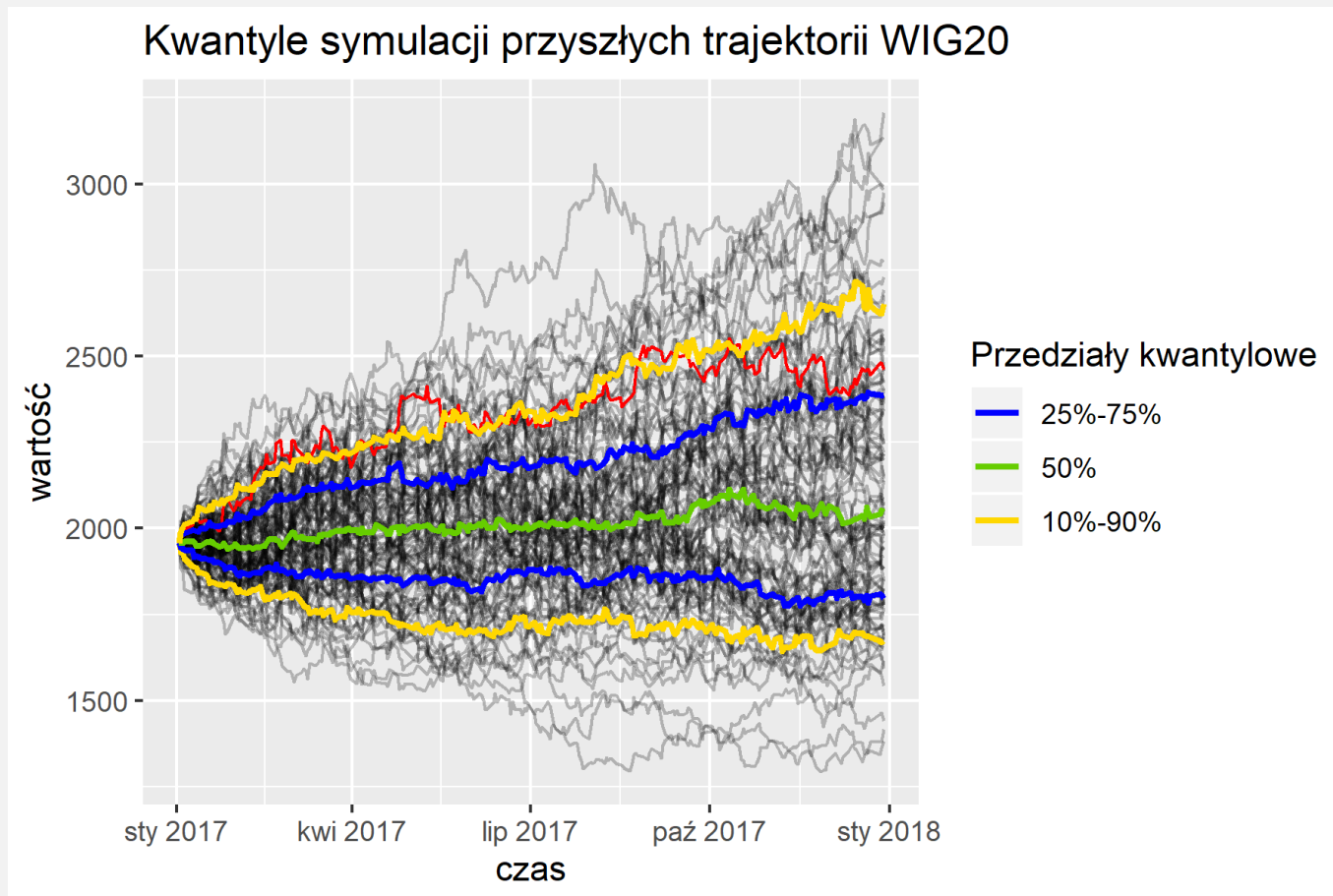
REZULTATY - WYKRESY

Symulacje przyszłych trajektorii:



REZULTATY - WYKRESY

Kwantyle symulacji
przyszłych trajektorii:



WSKAŹNIKI
GRECKIE – TABELA
TEORETYCZNA

	Call	Put
Value V Black–Scholes value	$Se^{-D(T-t)}N(d_1)$ $-Ee^{-r(T-t)}N(d_2)$	$-Se^{-D(T-t)}N(-d_1)$ $+Ee^{-r(T-t)}N(-d_2)$
Delta $\frac{\partial V}{\partial S}$ Sensitivity to underlying	$e^{-D(T-t)}N(d_1)$	$e^{-D(T-t)}(N(d_1) - 1)$
Gamma $\frac{\partial^2 V}{\partial S^2}$ Sensitivity of delta to underlying	$\frac{e^{-D(T-t)}N'(d_1)}{\sigma S\sqrt{T-t}}$	$\frac{e^{-D(T-t)}N'(d_1)}{\sigma S\sqrt{T-t}}$
Theta $\frac{\partial V}{\partial t}$ Sensitivity to time	$-\frac{\sigma Se^{-D(T-t)}N'(d_1)}{2\sqrt{T-t}}$ $+DSN(d_1)e^{-D(T-t)}$ $-rEe^{-r(T-t)}N(d_2)$	$-\frac{\sigma Se^{-D(T-t)}N'(-d_1)}{2\sqrt{T-t}}$ $-DSN(-d_1)e^{-D(T-t)}$ $+rEe^{-r(T-t)}N(-d_2)$
Speed $\frac{\partial^3 V}{\partial S^3}$ Sensitivity of gamma to underlying	$-\frac{e^{-D(T-t)}N'(d_1)}{\sigma^2 S^2(T-t)} \times$ $(d_1 + \sigma\sqrt{T-t})$	$-\frac{e^{-D(T-t)}N'(d_1)}{\sigma^2 S^2(T-t)} \times$ $(d_1 + \sigma\sqrt{T-t})$
Vega $\frac{\partial V}{\partial \sigma}$ Sensitivity to volatility	$S\sqrt{T-t}e^{-D(T-t)}N'(d_1)$	$S\sqrt{T-t}e^{-D(T-t)}N'(d_1)$
Rho (r) $\frac{\partial V}{\partial r}$ Sensitivity to interest rate	$E(T-t)e^{-r(T-t)}N(d_2)$	$-E(T-t)e^{-r(T-t)}N(-d_2)$
Rho (D) $\frac{\partial V}{\partial D}$ Sensitivity to dividend yield	$-(T-t)Se^{-D(T-t)}N(d_1)$	$(T-t)Se^{-D(T-t)}N(-d_1)$

$$d_1 = \frac{\log(S/E) + (r - D + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = \frac{\log(S/E) + (r - D - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} = d_1 - \sigma\sqrt{T-t}$$

WSKAŹNIKI GRECKIE – FUNKCJE W R

```

91  ### Giełdowe wskaźniki greckie
92
93  d1 <- function(spot, strike, r, sigma, t, T_){
94    asd = log(spot / strike) + (r + sigma * sigma / 2) * (T_ - t)
95    asd = asd / (sigma * sqrt(T_ - t))
96    asd
97  }
98  d2 <- function(spot, strike, r, sigma, t, T_){
99    return (d1(spot, strike, r, sigma, t, T_) - sigma * sqrt(T_ - t))
100 }
101
102 Delta <- function(spot, strike, r, sigma, t, T_, payoffType){
103   if(payoffType == 'call'){
104     return(pnorm(d1(spot, strike, r, sigma, t, T_)))
105   }
106   if(payoffType == 'put'){
107     return(vanillaDelta(spot, strike, r, sigma, t, T_, 'call') - 1)
108   }
109 }
110
111 Gamma <-function(spot, strike, r, sigma, t, T_, payoffType){
112   if(payoffType == 'call'){
113     return(dnorm(d2(spot, strike, r, sigma, t, T_))/(spot * sigma * sqrt(T_ - t)))
114   }
115 }
116 Theta <-function(spot, strike, r, sigma, t, T_, payoffType){
117   d1 = d1(spot, strike, r, sigma, t, T_)
118   d2 = d2(spot, strike, r, sigma, t, T_)
119   if(payoffType == 'call'){
120     return(-sigma * spot * pnorm(d1)/(2 * sqrt(T_ - t))-r * strike* exp(-(T_-t)*r)*dnorm(d2))
121   }
122   if(payoffType == 'put'){
123     return(-sigma * spot * pnorm(-d1)/(2 * sqrt(T_ - t))+r * strike* exp(-(T_-t)*r)*dnorm(-d2))
124   }
125 }
126 Vega <-function(spot, strike, r, sigma, t, T_, payoffType){
127   return(spot * sqrt(T_ - t) * pnorm(d1))
128 }
129 Rho <-function(spot, strike, r, sigma, t, T_, payoffType){
130   if(payoffType == 'call'){
131     return(spot * (T_ - t) * sigma * exp(-(T_-t)*r) * dnorm(d2))
132   }
133   if(payoffType == 'put'){
134     return(-spot * (T_ - t) * sigma * exp(-(T_-t)*r) * dnorm(-d2))
135   }
136 }
137 #cena
138 vanillaPrice <- function(spot, strike, r, sigma, t, T_, payoffType){
139   discountFactor = exp(-r * (T_ - t))
140   d1 = d1(spot, strike, r, sigma, t, T_)
141   d2 = d2(spot, strike, r, sigma, t, T_)
142   if(payoffType == 'call'){
143     return(spot * pnorm(d1) - strike * discountFactor * pnorm(d2))
144   }
145   if(payoffType == 'put'){
146     return(- spot * pnorm(-d1) + strike * discountFactor * pnorm(-d2))
147   }
148 }

```

KONIEC