Pracovní text a úkoly ke cvičením MF002

Ondřej Pokora, PřF MU, Brno

11. března 2013

1 Brownův pohyb (Wienerův proces)

Základním stavebním kamenem simulací náhodných procesů popsaných pomocí stochastických diferenciálních rovnic je generování trajektorií Brownova pohybu (Wienerova procesu) $W_t = W(\omega, t)$.

Definice říká, že počáteční hodnotou je $W_0 = 0$, ale s jakoukoliv jinou známou hodnotou se pracuje podobně. Přírůstky Brownova pohybu za časové intervaly délky Δt jsou náhodné veličiny $\Delta W \sim \mathrm{N}(0, \Delta t)$, a pro disjunktní časové intervaly jsou tyto přírůstky stochasticky nezávislé.

Vlastní simulace trajektorie Brownova pohybu spočívá v dělení požadovaného časového intervalu $[0, T]$ na velmi krátké podintervaly délek Δt. V těchto jemně zvolených časech postupně (s rostoucím časem) počítáme hodnotu Brownova pohybu tak, že k hodnotě v bezprostředně předchozím známém čase připočítáváme realizaci náhodného přírůstku ΔW.

Pro hodnotu $W(\omega, t + \Delta t)$ tedy platí:

$$W(\omega, t + \Delta t) = W(\omega, t) + \Delta W \ ,$$

přičemž $\Delta W = \sqrt{\Delta t}\, U$, kde U je realizace náhodné veličiny s rozdělením $\mathrm{N}(0, 1)$.

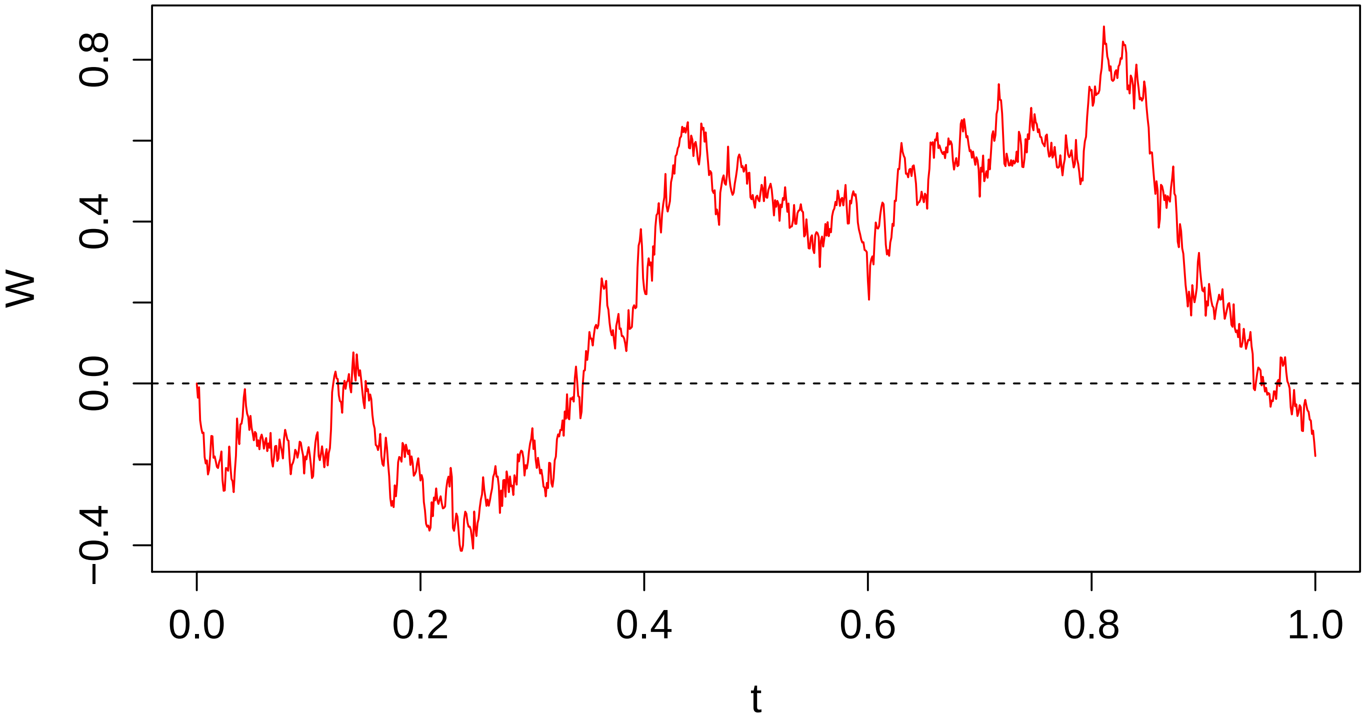
Projděte si následující skript, který W_t generuje pomocí **for**-cyklu:

```
W0 <- 0
dt <- 0.001
Wt <- W0
W <- W0
t <- seq (0, 1, by=dt)

for (k in 1:1000) {
  dW <- sqrt (dt) * rnorm (1)
  Wt <- Wt + dW
  W <- append (W, Wt)
}

plot (t, W, type="l", col="red", xlab="t", ylab="W")
abline (h = W0, lty = 2)
```

Výše uvedený postup se v moderních matematických programech (*R, Matlab*) provádějících vektorové (maticové) operace nepoužívá.

Místo toho se jen jedním průchodem generuje celý náhodný výběr (vektor) přírůstků (vizte nezávislost z definice W_t), který se kumulativně přičítá (funkcí **cumsum**) k výchozí hodnotě W_0:

```
W0 <- 0
dt <- 0.001
W <- W0
t <- seq (0, 1, by=dt)

dW <- rnorm (length (t) - 1) * sqrt (dt)
W <- cumsum (c (W0, dW))

plot (t, W, type="l", col="red", xlab="t", ylab="W")
abline (h = W0, lty = 2)
```

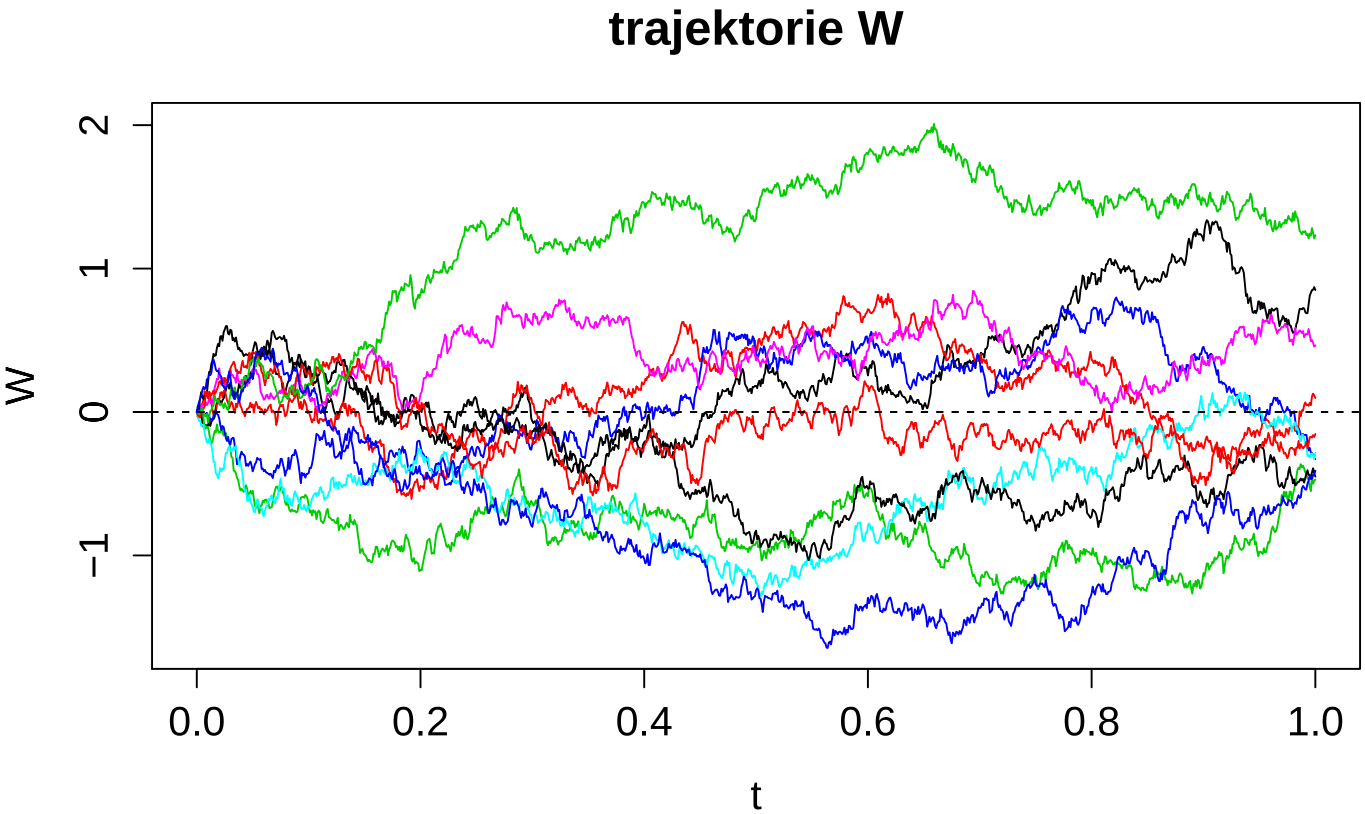


Dva řádky skriptu generující W_t je vhodné si uložit jako vlastní funkci. Následně totiž můžete jen změnou parametrů této funkce snadno generovat trajektorii:

```
generuj.Wp <- function (t, dt, W0) {
  dW <- rnorm (length (t) - 1) * sqrt (dt)
  W <- cumsum (c (W0, dW))
}

W <- generuj.Wp (t, dt, W0)

plot (t, W, type="l", col="red", xlab="t", ylab="W")
abline (h = W0, lty = 2)
```

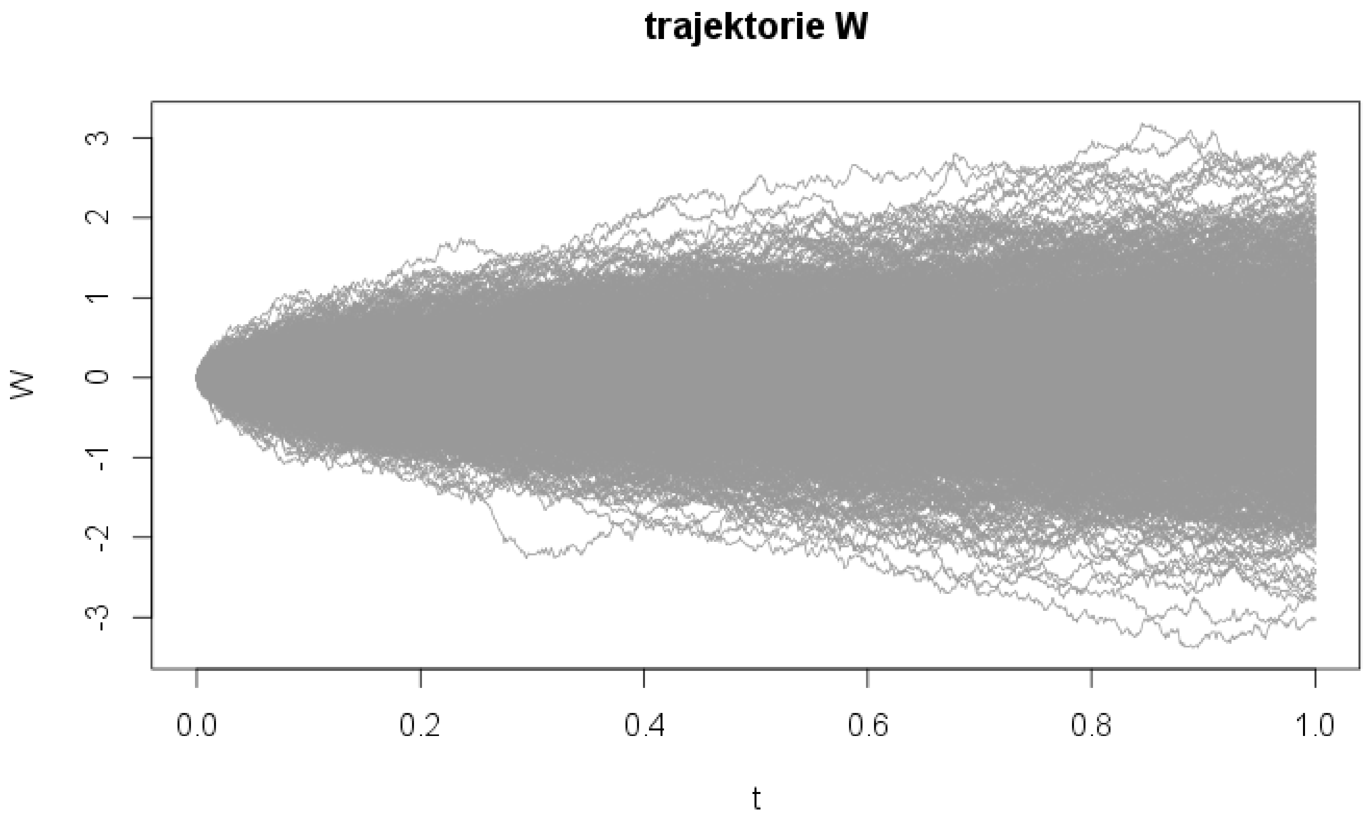
Síla funkce se ukáže především, chceme-li vygenerovat více trajektorií:

```
M <- sapply (1:10, function (k) {
  generuj.Wp (t, dt, W0)
})

matplot (t, M, type = "l", lty = 1, xlab = "t", ylab = "W", main="trajektorie W")
abline (h = W0, lty = 2)
```



Pouhou změnou rozsahu prvního parametru funkce **sapply** lze měnit počet trajektorií. Výsledné vektory hodnot pro jednotlivé trajektorie se přitom skládají vedle sebe, takže výsledná proměnná je matice, v níž každý sloupec přísluší jedné trajektorii a každý řádek odpovídá jednomu času pozorování.

Vyzkoušejte si vygenerovat 1000 (nebo 10000) trajektorií W_t. Vykreslete je do grafu, seznamte se s parametry příkazů **plot** a **matplot** pro rozsahy a popisy souřadnicových os, a pro volbu stylu, tloušťky, symbolu a barvy čar zobrazených dat. Měli byste dostat podobný graf:



Dále budeme zkoumat rozdělení pravděpodobnosti realizací W_t pro jeden konkrétní čas $t = 0.6$. Nejprve potřebujeme zjistit, který řádek tomuto času odpovídá.

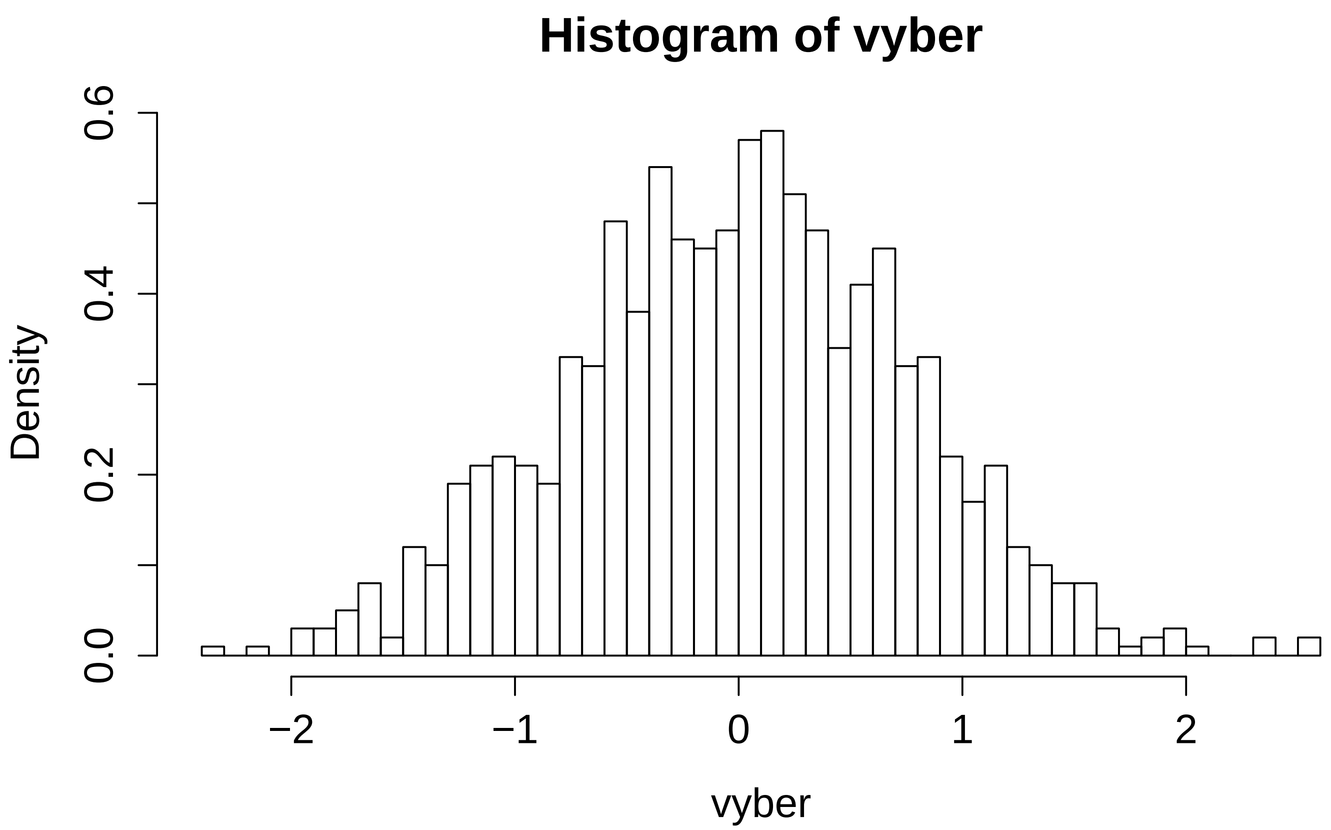
```
index <- which (t==0.6)
vyber <- M[index,]
```

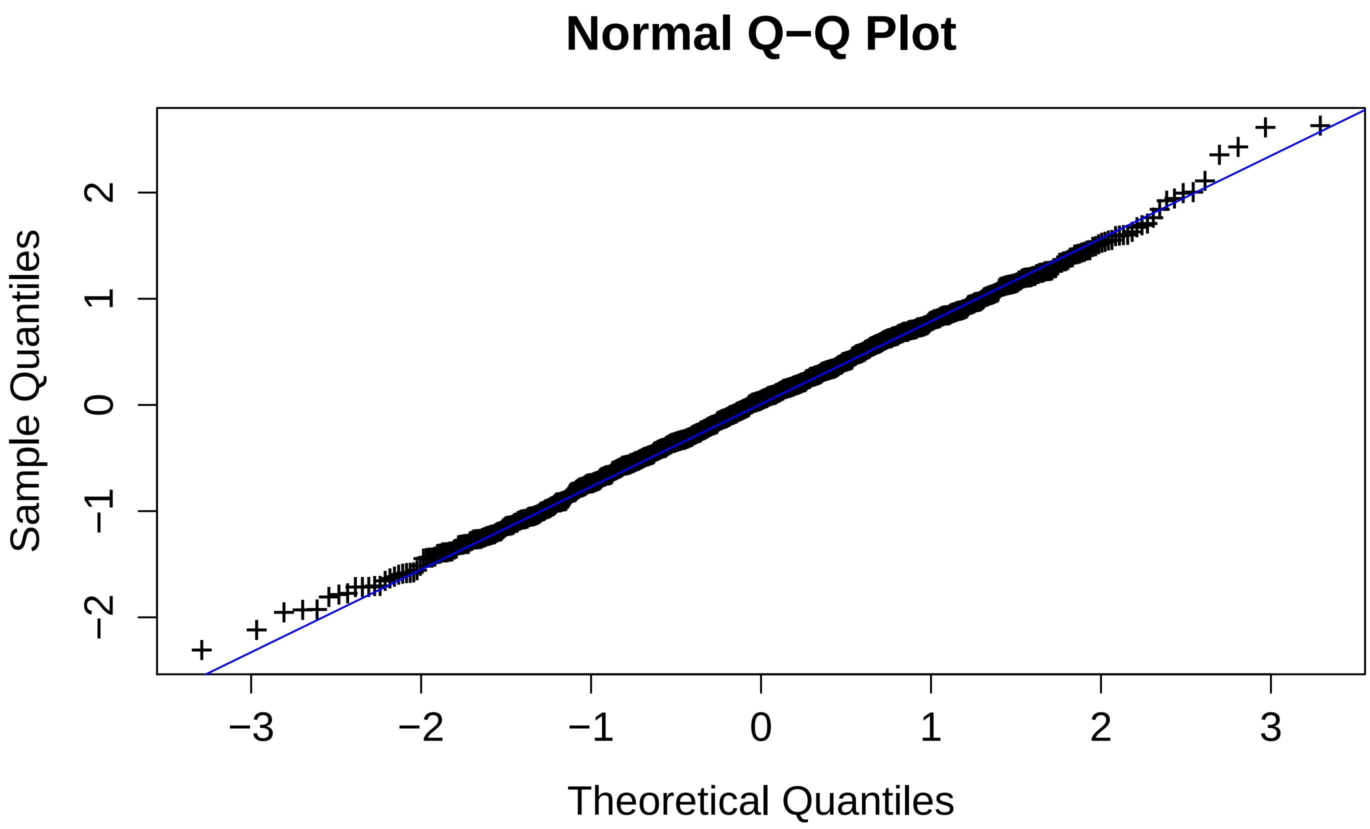
Vybrali jsme z matice trajektorií jen jeden řádek odpovídající požadovanému času $t = 0.6$, který je z pohledu statistiky náhodným výběrem.

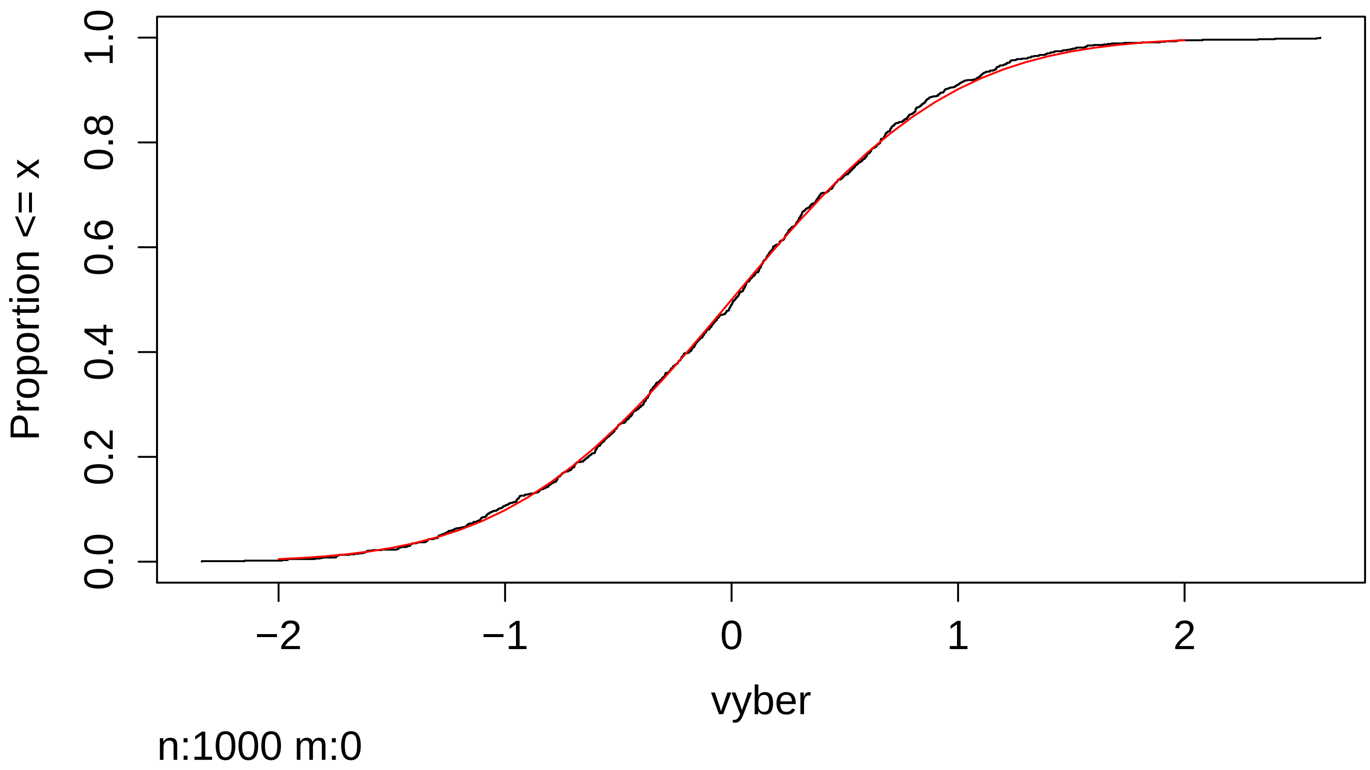
Následující příkazy shrnují nejčastěji používané nástroje pro ověřování rozdělení pravděpodobnosti: histogram, QQ-plot, empirickou distribuční funkci (ecdf) a Kolmogorovův-Smirnovův test. Pomocí nápovědy prozkoumejte použití příkazů a především interpretaci výsledků.

```
hist (vyber, breaks = 50, freq = FALSE)

qqnorm (vyber, pch = "+")
qqline (vyber, col = "blue")

library (Hmisc)
Ecdf (vyber)
x <- seq (-2, 2, by=0.1)
y <- pnorm (x, mean = 0, sd=sqrt (0.6))
lines (x, y, col = "red")

ks.test (vyber, pnorm, mean = 0, sd=sqrt (0.6))
ks.test (vyber, pnorm, mean = 1, sd=sqrt (0.6))
ks.test (vyber, pnorm, mean = 0, sd=1)
```

Měli byste dostat podobné grafy:



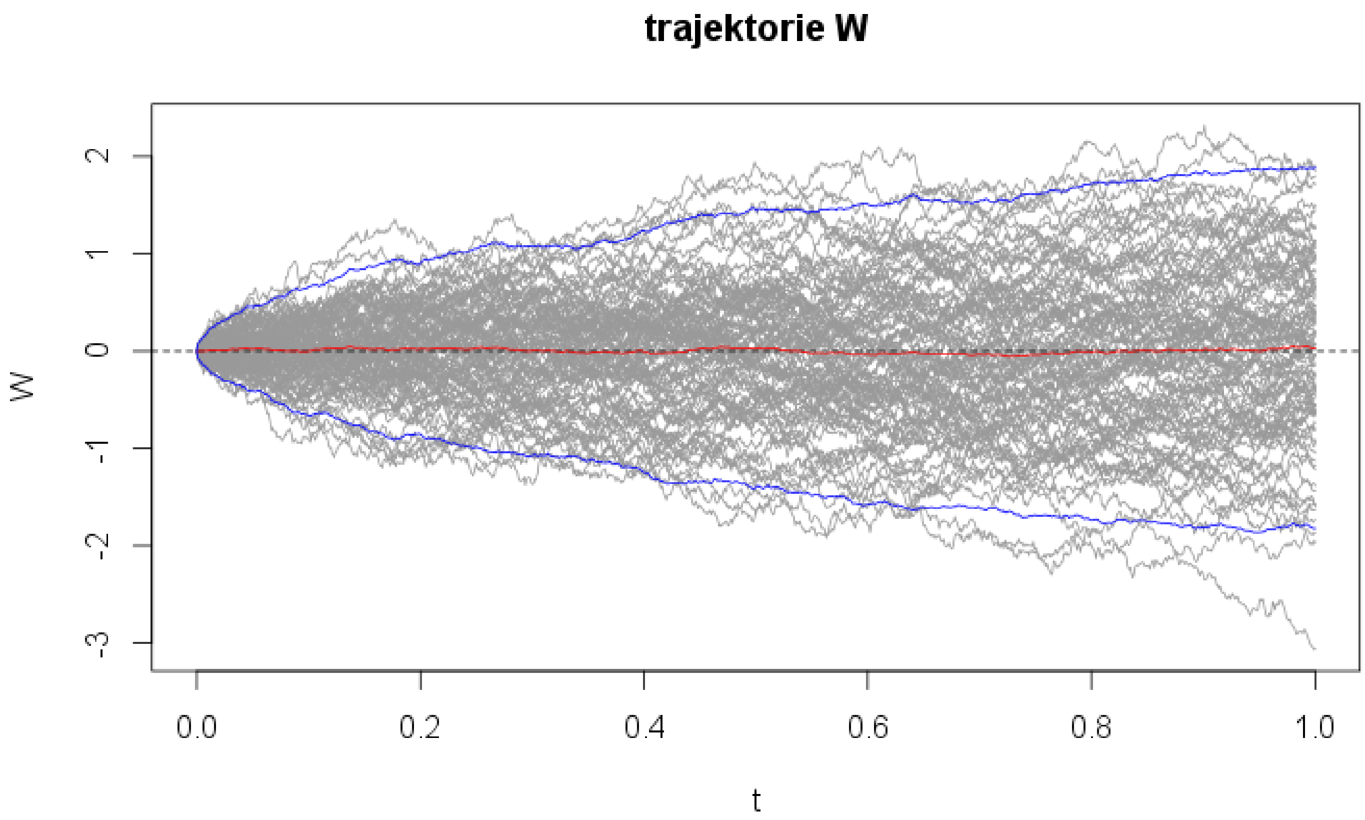


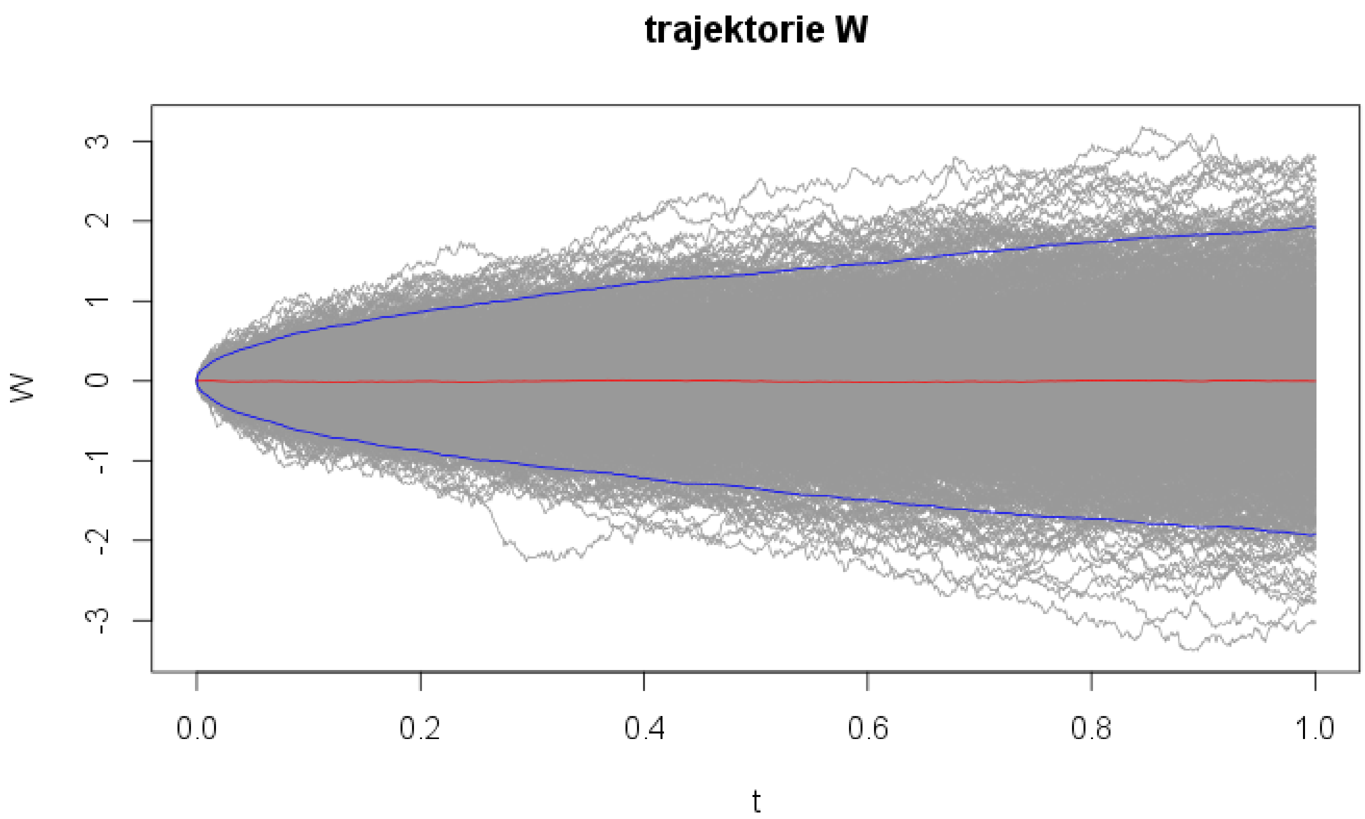


1.1 Úkoly

Generujte 10, 100, 1000, 10000 trajektorií W_t a vykreslete je do grafu příkazem **matplot**.

K matici realizací Wienerova procesu spočítejte střední hodnotu a směrodatnou odchylku pro každý čas (tedy pro každý řádek matice). K tomu se bude hodit funkce **apply** pro aplikaci na každý řádek či sloupec matice, a funkce **mean** pro střední hodnotu a **sd** pro směrodatnou odchylku. Do obrázku trajektorií pak přidejte křivku pro střední hodnoty a 95% interval spolehlivosti realizací. Měli byste dostat obrázky podobné následujícím, které jsou pro 100 a 1000 trajektorií:





Pro jednotlivé varianty (10, 100, 1000, 10000 trajektorií) si pro realizace Wienerova procesu v časech $t = 0.2$ a $t = 0.8$ zobrazte histogram a QQ-plot. Vykreslete si také empirickou distribuční funkci a graficky ji porovnejte s teoretickou distribuční funkcí. Jaké rozdělení pravděpodobnosti a s jakými parametry má W_t pro $t = 0.2$ a $t = 0.8$?

Definujme náhodné procesy

$$Y(\omega, t) = \min_{s \leq t} W(\omega, s) \ ,$$

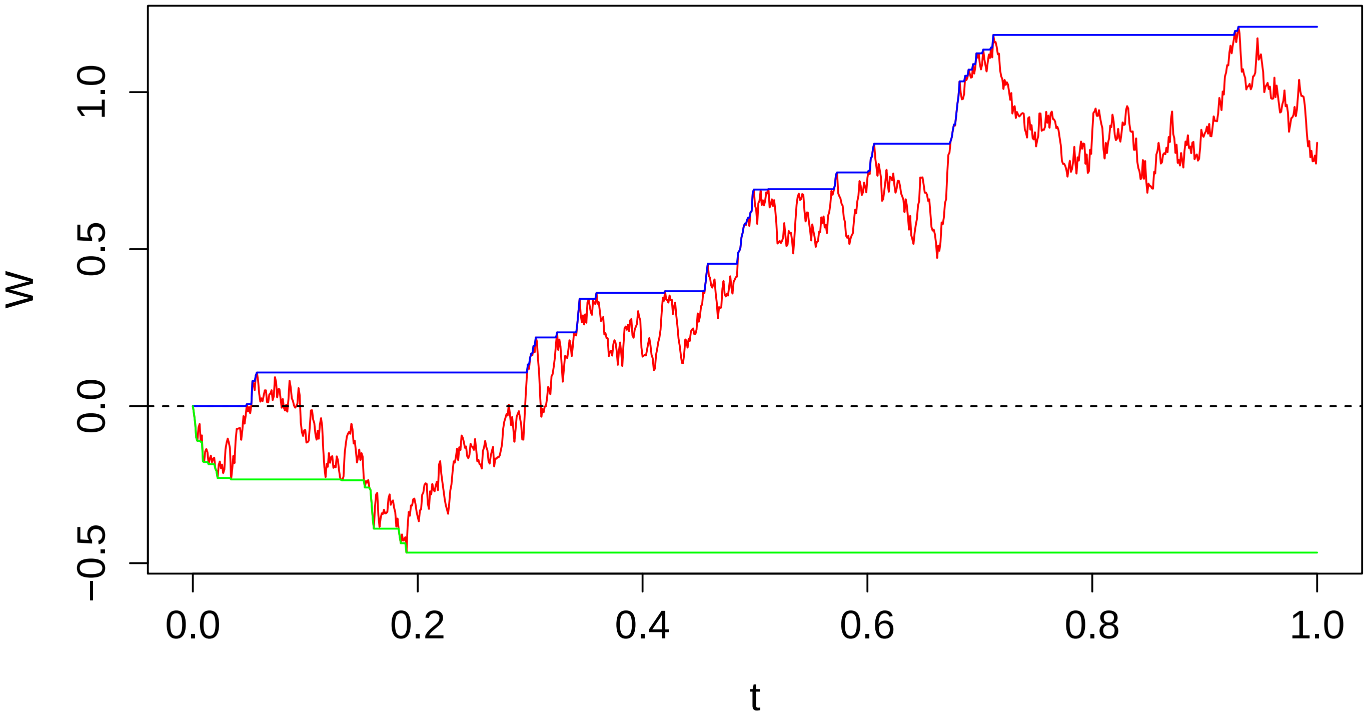
a

$$Z(\omega, t) = \max_{s \leq t} W(\omega, s) \ ,$$

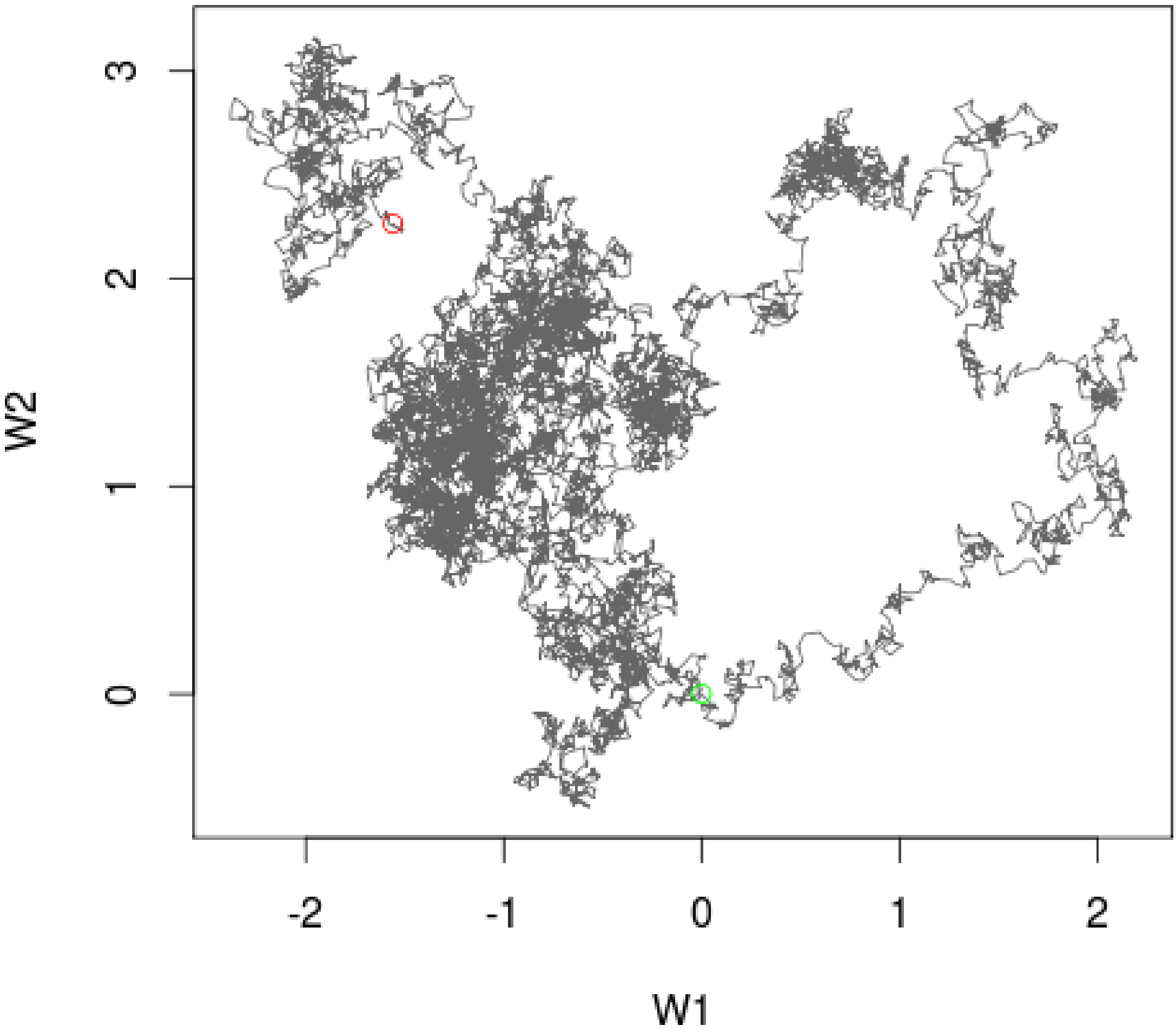
tzn. minimum a maximum Wienerova procesu na intervalu $[0, t]$. Vygenerujte si jednu trajektorii Wienerova procesu W_t na intervalu $[0, 1]$ a k ní spočítejte trajektorie Y_t, Z_t na stejném intervalu. Pomohou funkce **min** a **max**, a výběr podvektoru pomocí hranatých závorek, např.

```
max (W[1:50])
```

spočítá maximum z prvních 50 hodnot trajektorie Wienerova procesu. Trajektorie všech tří procesů pak zobrazte v jednom grafu:



Vytvořte obrázek 2rozměrného Brownova pohybu. Každá ze dvou souřadnic je tvořena Wienerovým procesem. Generujte tedy 2 Wienerovy procesy pro časový interval $[0, 10]$ s krokem $\Delta t = 0.001$ a pomocí příkazu **plot** každý použijte jako souřadnice na jiné ose.



W2
3
2
1
0
-2
-1
0
1
2
W1