

Mathematica, R a SQL

Jakub Černý
KPMS MFF UK
jcerny@karlin.mff.cuni.cz

ROBUST 19-24.1 .2014, Jetřichovice

Obsah

- **Úvod**
- **Propojení softwaru Mathematica 9 s programem R (RLink)**
 - R funkce v Mathematice (RSet, REvaluate, RFunction)
 - Praktický příklad - Shluková analýza
- **Mathematica 9 a SQL (DatabaseLink)**
 - SQL funkce v Mathematice (SQLSelect, SQLInsert, SQLUpdate, SQLDelete, SQLExecute)
- **Generování procesů**
 - Obecně a konkrétní difúzní procesy

Úvod

Užitečné informace o systému Mathematica

- Mathematica není pouze výpočetní systém, ale i prostředí pro prezentace, publikace, vizualizace, výpočty i vytváření aplikací.
- Velkou výhodou Mathematicy, oproti ostatním podobným programům, jsou symbolické výpočty, např. derivace hustoty normálního rozdělení dle parametrů

$\partial_{\alpha} \text{PDF}[\text{NormalDistribution}[\alpha, \beta], \mathbf{x}]$

$$\frac{e^{-\frac{(x-\alpha)^2}{2\beta^2}} (x-\alpha)}{\sqrt{2\pi} \beta^3}$$

$\partial_{\beta} \text{PDF}[\text{NormalDistribution}[\alpha, \beta], \mathbf{x}]$

$$\frac{e^{-\frac{(x-\alpha)^2}{2\beta^2}} (x-\alpha)^2}{\sqrt{2\pi} \beta^4} - \frac{e^{-\frac{(x-\alpha)^2}{2\beta^2}}}{\sqrt{2\pi} \beta^2}$$

- Umožňuje připojení k rozsáhlým veřejným zdrojům dat, např. CountryData - data o zemích, FinancialData - finanční data,... (vyžaduje připojení k Internetu)

`CountryData["Czech Republic", #] & /@`

`{"Population", "GDP", "NaturalGasConsumption", "HIVAIDSPopulation"}`

`{1.0411 × 107, 2.17077 × 1011, 8.719 × 109, 1500.}`

- Nabízí intuitivní zadávání příkazů a intuitivní vyhledávání dat

 `$Aborted`

$$\frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi} \sigma}$$

- Poskytuje přístup k různým datům pomocí WolframAlpha

 **Czech Republic Unemployment rate development 2005-2012**

Input interpretation: +

Czech Republic	unemployment rate	2005 to 2012
----------------	-------------------	--------------

[Definitions](#)

Result: +

mean	6.55%
lowest	4.4% (2008)
highest	7.9% (2005)

(2005 to 2012)

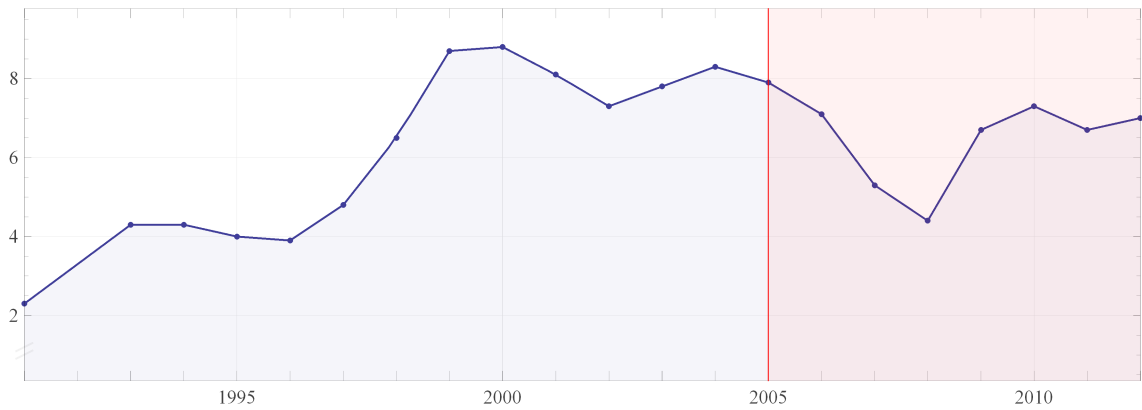
(2000 to 2012)

(fraction of total labor force)

Unemployment rate history:

Log scale

All years



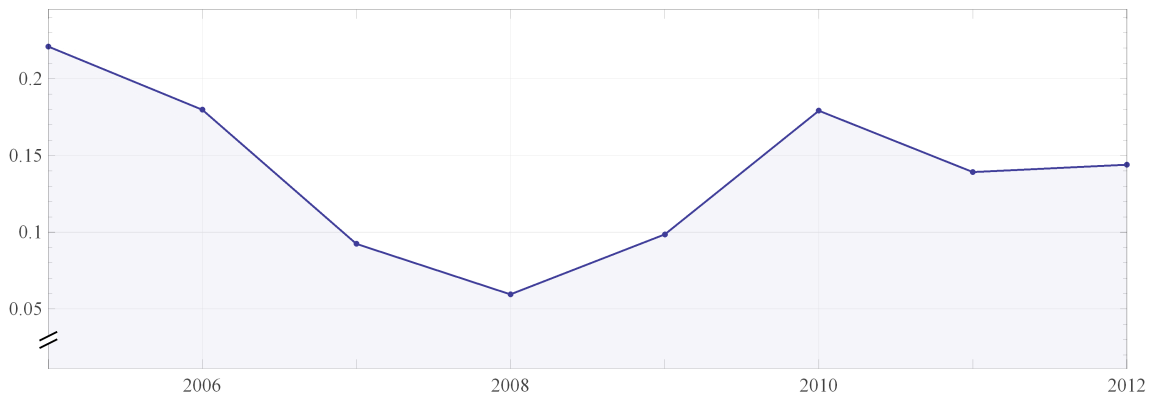
(from 1991 to 2012) (in percent)

(fraction of total labor force)

Long-term unemployment rate:

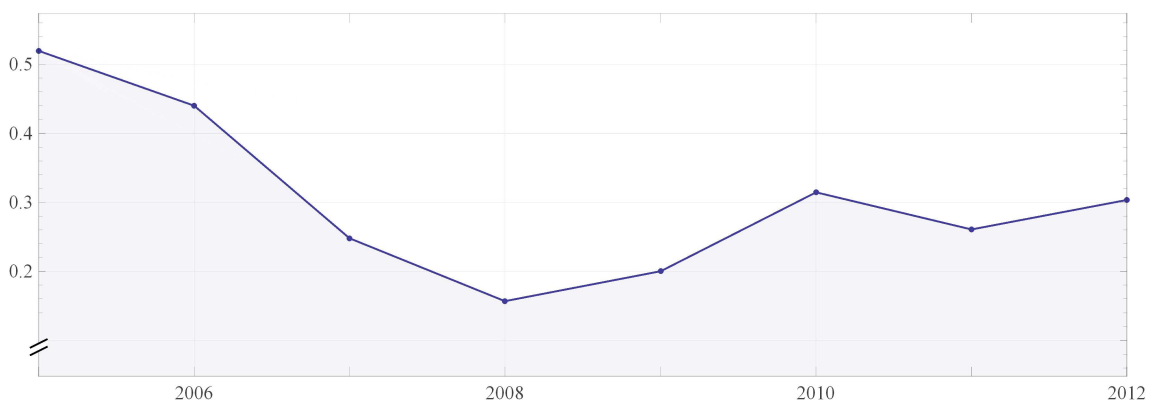


Male:



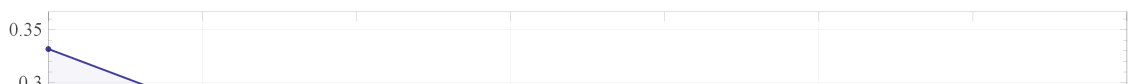
(from 2005 to 2012) (in percent)

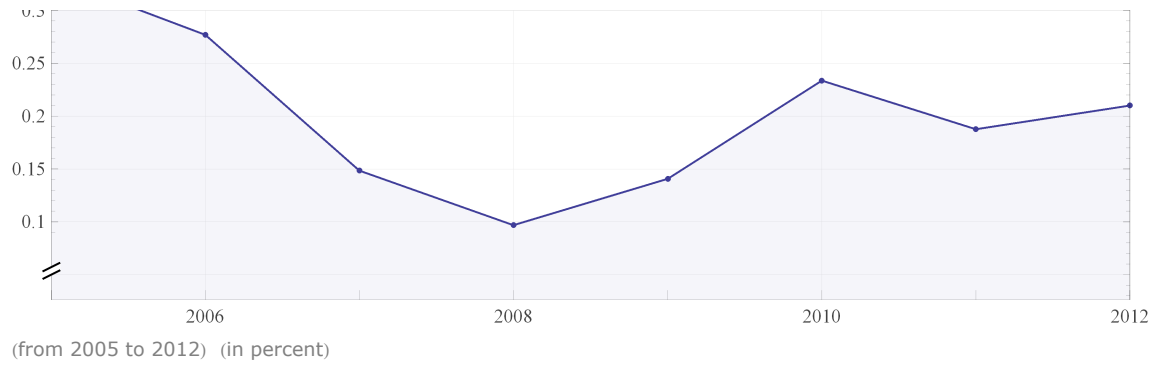
Female:



(from 2005 to 2012) (in percent)

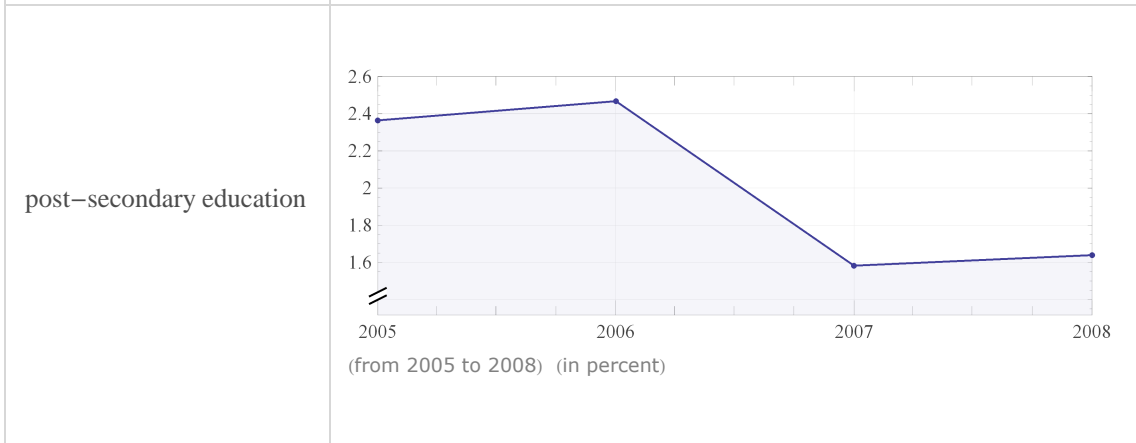
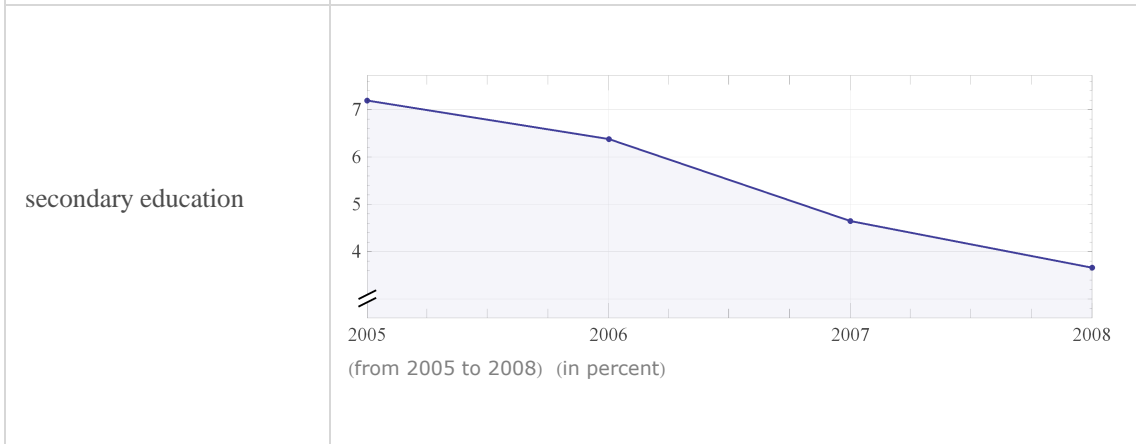
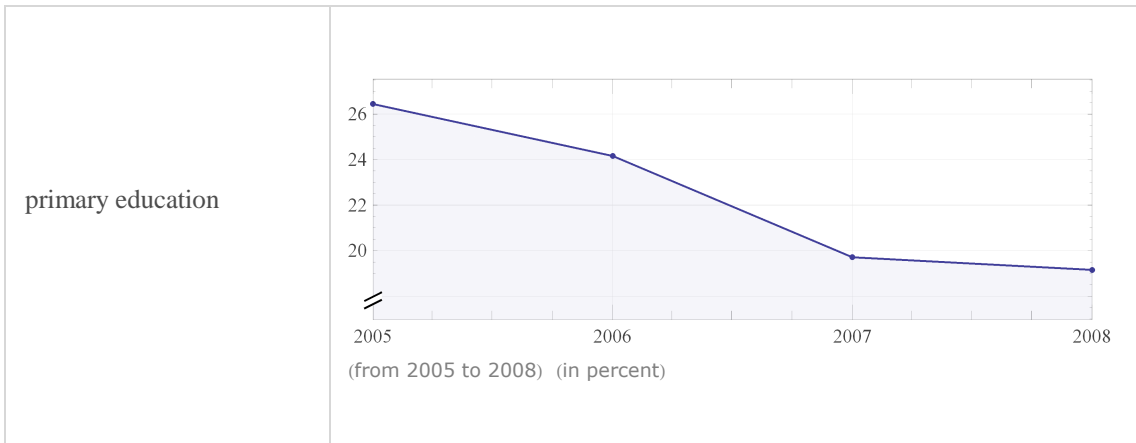
Total:



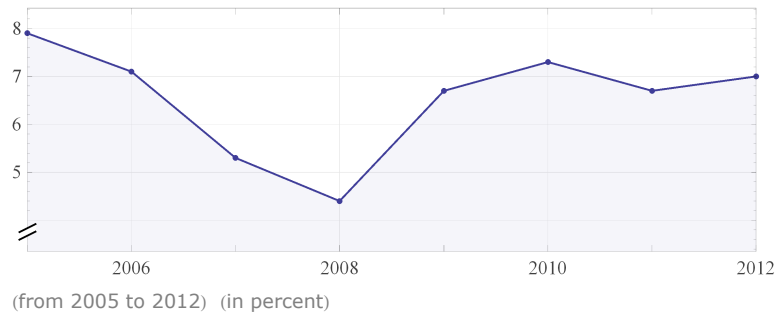


Unemployment rate by education:

[Show details](#) +



total



[Definitions](#)

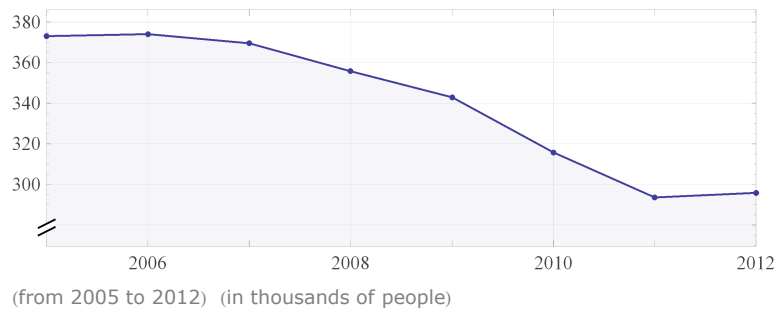
Labor force by education:

[Show details](#)

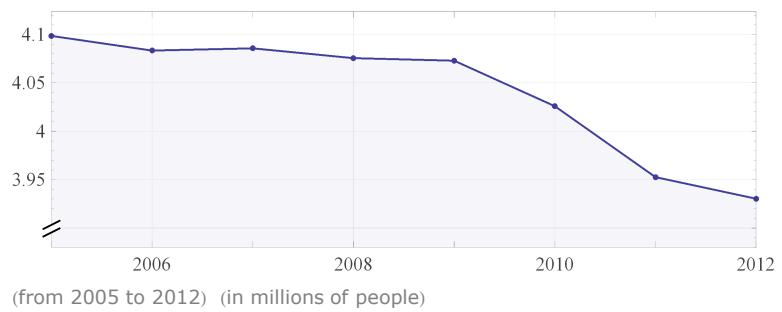
[Show chart](#)



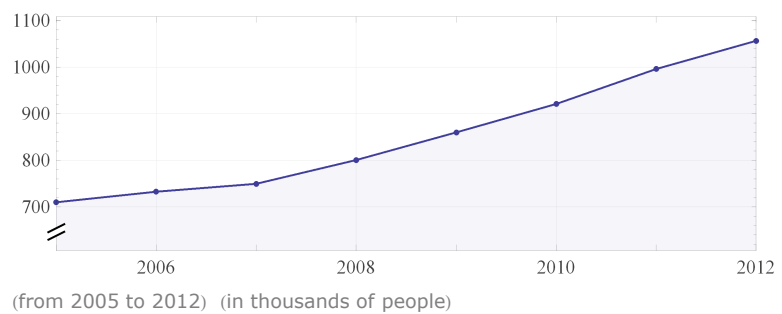
primary education

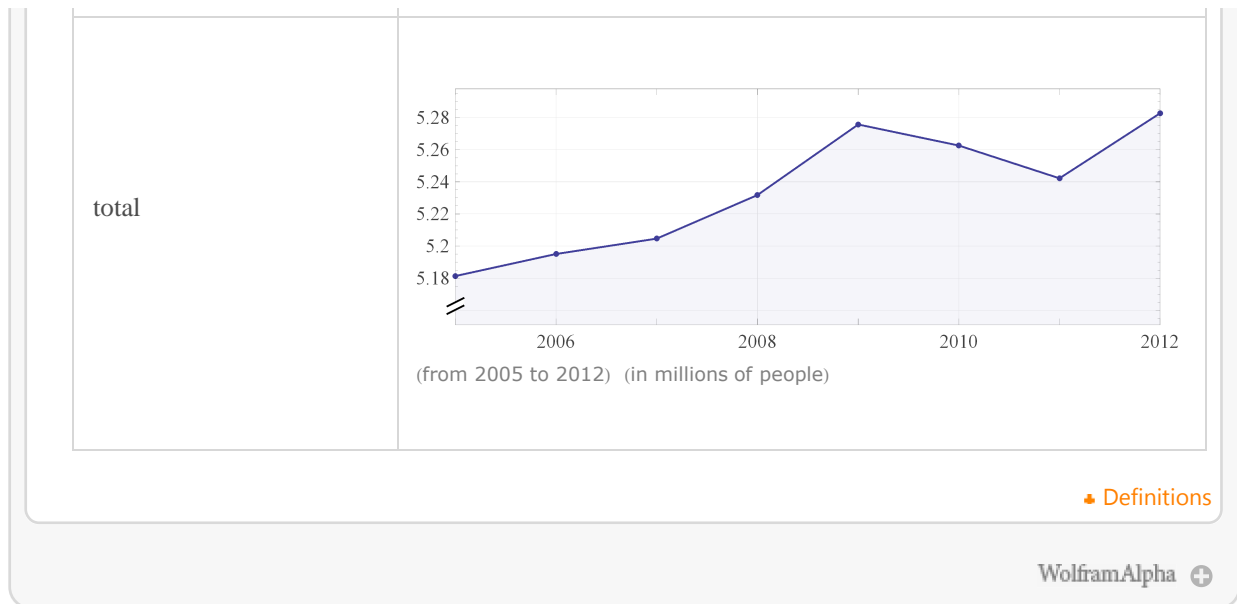


secondary education



post-secondary education





Zajímavé novinky ve verzi 9

■ Propojení Mathematicy a externího softwaru

- *MathLink* - propojení např. s C++
- *LibraryLink* - nahrání a kompilování dynamických knihoven (.dll)
- *JLink* - propojení s Javou (vytváření GUI)
- *NETLink* - propojení s .NET
- *DatabaseLink* - propojení s databázemi, např. SQL, MS Access,...
- *RLink* - propojení s programem R

■ Časové řady a stochastické diferenciální rovnice

- *ARProcess*, *MAProcess*, *ARMAProcess* - autoregresní proces a proces klouzavých průměrů
- *ItoProcess*, *StratonovichProcess*, *GeometricBrownianMotionProcess*, *OrnsteinUhlenbeckProcess*, *BrownianBridgeProcess* - obecný Itoův a Stratonovichův proces a vybrané konkrétní procesy (geometrický Brownův pohyb, Ornstein - Uhlenbeck proces a Brownův most)
- *EstimatedProcess* - funkce, která odhadne parametry procesu z dat
- *SliceDistribution* - funkce, která určí rozdělení procesu v obecném čase t

■ Náhodné procesy

- *QueueingProcess*, *QueueingNetworkProcess* - procesy hromadné obsluhy, Jacksonova síť
- *StationaryDistribution* - funkce, která nalezne stacionární rozdělení, pokud existuje

a mnoho dalších.

Propojení softwaru *Mathematica* 9 s programem R (*RLink*)

- Propojit program R s Mathematicou lze pouze ve verzi *Mathematica* 9 a vyš. Ke spuštění je zapotřebí balík **RLink** a nainstalování Rka do Mathematicy.

```
Needs["RLink`"]
```

```
InstallR[]
```

- Funkce **InstallR** může mít i další argumenty, pokud chceme změnit následující možnosti, tj. např. cesta k JRE, cesta k externím definicím datových typů, apod.

```
Options[InstallR]
```

```
{JRELocation -> Automatic, TargetPlatform -> Automatic,  
RCommandLine -> Automatic, AddToRDataTypePath -> None,  
RHomeLocation -> Automatic, EnableResourcesAutoinstall -> True}
```


Funkce RSet

- Funkcí **RSet** se přiřadí konstanty, vektory a matice z Mathematicy do Rkové proměnné, která musí být uvedena v uvozovkách. Např.

- konstantu

```
RSet["x", 100]
```

```
100
```

- hodnotu napočítanou v Mathematice (vestavěná funkce s číselným výstupem)

```
val1 = N[Exp[-10]]
```

```
RSet["x1", val1]
```

```
0.0000453999
```

```
0.0000453999
```

- hodnotu napočítanou v Mathematice (vlastní funkce s číselným výstupem)

```
funkce[a_, b_] := (a + b) / 2
```

```
RSet["x2", funkce[1, 2.]]
```

```
1.5
```

- vektor

```
RSet["x3", Range[10]]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

- matici

```
RSet["x4", {{1, 0}, {0, 1}}]
```

```
{{1, 0}, {0, 1}}
```

- přiřazení hodnoty do vektoru v R

```
RSet["vektor", {1, 2, 3, 4}]
```

```
RSet["vektor[[1]]", 4]
```

```
{1, 2, 3, 4}
```

```
4
```

```
REvaluate["vektor"]
```

```
{4, 2, 3, 4}
```

- hodnota dimenze

```
RSet["vektor", Range[12]]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
```

```
RSet["dim(vektor)", {2, 6}]
```

```
{2, 6}
```

```

REvaluate["vektor"] // MatrixForm

$$\begin{pmatrix} 1 & 3 & 5 & 7 & 9 & 11 \\ 2 & 4 & 6 & 8 & 10 & 12 \end{pmatrix}$$


```

- Funkce, které mají výsledek v symbolickém tvaru (funkce Exp), nebo hodnoty, které mají symbolický zápis (číslo π) nelze do Rkové proměnné přiřadit

```
Exp[-1]
```

$$\frac{1}{e}$$

```
RSet["x3", Exp[-1]]
```

RSet::badval : The expression $\frac{1}{e}$ is not convertible to R >>

```
$Failed
```

```
Pi
```

$$\pi$$

```
RSet["x4", Pi]
```

RSet::badval : The expression π is not convertible to R >>

```
$Failed
```

Funkce *REvaluate*

- Funkce **REvaluate** vyhodnotí libovolný Rkový kód. Kód je argumentem funkce **REvaluate** a musí být vždy celý v uvozovkách. Pokud se má vyhodnotit více než jeden příkaz, musí být kód ve složených závorkách. Např.

```
x = REvaluate["1:100"]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62,
 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100}
```

```
REvaluate["matice <- array(1:9,dim=c(3,3))"] // MatrixForm
```

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

```
REvaluate["{
  vlcisla <- eigen(matice)
  vlcisla
}"][[1]]
```

```
{{16.1168, -1.11684, -5.70069 × 10-16}, {-0.464547, -0.882906, 0.408248},
{-0.570796, -0.23952, -0.816497}, {-0.677044, 0.403865, 0.408248}}
```

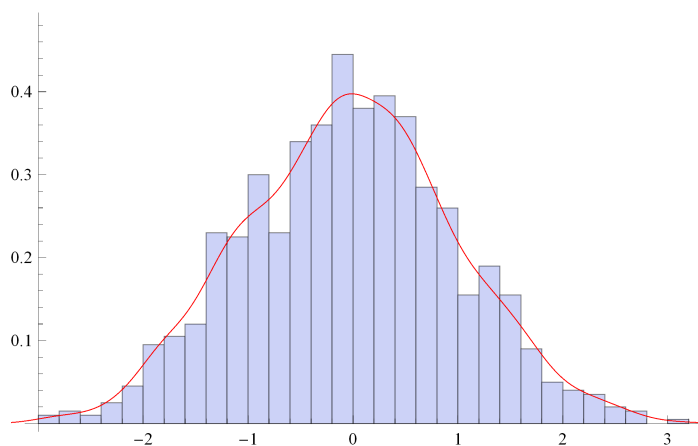
- Pokud jsou ve funkci **REvaluate** vytvářeny nové proměnné, **REvaluate** hodnotu vrací poslední z nich

```
nahCisla = REvaluate["
```

```
{
  a <- 1000
  b <- 0
  c <- 1
```

```
vystup <- rnorm(a,b,c)
}"];
```

```
Show[Histogram[nahCisla, Automatic, "PDF"],
SmoothHistogram[nahCisla, Automatic, "PDF", PlotStyle → Red]]
```



- Samotné uvozovky (kromě začátku a konce) v kódu R být nemůžou, používá se zde zápis stejný jako v TeXu

```
REvaluate["{
  hist(vystup,prob = T, main = \"Graf č. 1\")
  lines(density(vystup),col = 2)
}"]
```

- Mathematica stáhne libovolnou knihovnu, která je v kódu volána.

```
REvaluate["{
  library(MASS)
  mvrnorm(10,c(0,0),matrix(c(1,0.5,0.5,1),nrow = 2, ncol = 2))
}"]
```

```
RObject[{{{-1.57046, -0.785324}, {-1.38512, 0.0986001}, {-0.337468, 0.415469},
  {0.191569, 0.259396}, {-0.0675553, 1.28605}, {0.500687, 0.412701},
  {-0.145979, 1.05609}, {0.14002, -0.73727}, {1.45772, 3.01487},
  {-0.788201, -0.511074}}, RAttributes[dimnames -> {Null, Null}]]
```

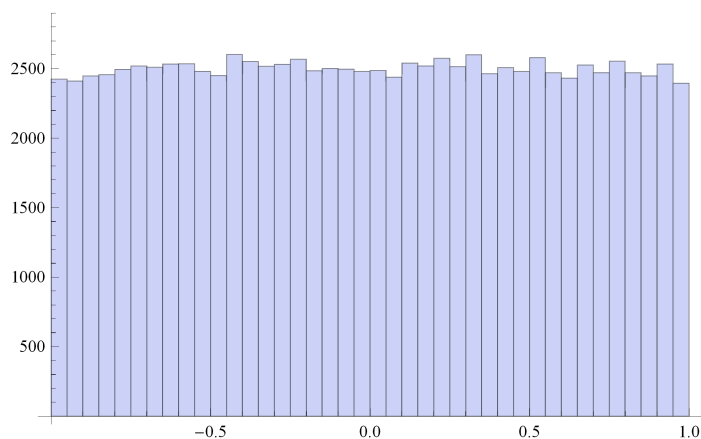
```
REvaluate["{
  library(MASS)
  mvrnorm(10,c(0,0),matrix(c(1,0.5,0.5,1),nrow = 2, ncol = 2))
}"][[1]]
{{{-1.7984, 0.849506}, {0.911511, 0.958298},
  {1.90097, 1.61864}, {-2.16663, -1.81885}, {-0.693929, 0.0521931},
  {0.182485, -0.875293}, {-0.139312, -1.04394},
  {1.09479, 0.247195}, {1.18746, -0.33308}, {0.56189, -0.652747}}}
```

Funkce *RFunction*

- Funkce **RFunction** slouží k vyhodnocení vlastních Rkových funkcí v Mathematice, pomocí této funkce se také dají předávat hodnoty z programu R do Mathematicy

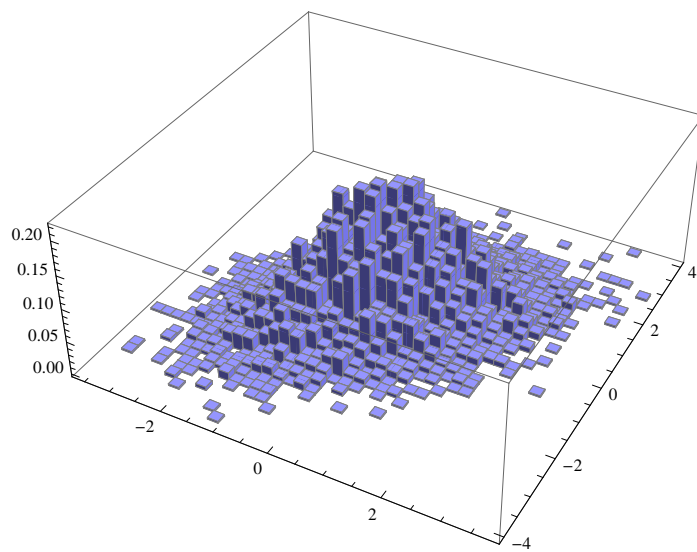
```
Clear[nahCisla];
nahCisla = RFunction["function(a,b,c) {
  vystup <- runif(a,b,c)
}"]
RFunction[closure, RCode[function(a,b,c) {
  vystup <- runif(a,b,c)
}], Automatic, RAttributes[]]
```

```
Histogram[nahCisla[100 000, -1, 1]]
```



```
Clear[nahVektor];
nahVektor = RFunction["function(pocet) {
  library(MASS)
  mvrnorm(pocet,c(0,0),matrix(c(1,0.5,0.5,1),nrow = 2, ncol = 2))
}"]
RFunction[closure, RCode[function(pocet) {
  library(MASS)
  mvrnorm(pocet,c(0,0),matrix(c(1,0.5,0.5,1),nrow = 2, ncol = 2))
}], Automatic, RAttributes[]]
```

```
Histogram3D[nahVektor[10 000][[1]], Automatic, "PDF"]
```



Praktický příklad - Shluková analýza, Kmeans

Dow Jones Industrial Average (DJIA)

- Dow Jones Industrial Average (DJIA), označován často jako **Dow Jonesův index**, je jeden z neznámějších a nejstarších ukazatelů vývoje na americkém akciovém trhu.
- DJIA sleduje akcie 30 amerických společností, které patří mezi největší a nejvíce obchodované na New York Stock Exchange (NYSE). Od roku 1896, kdy byl poprvé publikován se jeho struktura značně změnila. Jediná společnost, která je v indexu od jeho založení, je společnost General Electric
- Způsob výpočtu DJIA :

$$DJIA_t = \frac{1}{D_t} \sum_{i=1}^{30} p_t^i,$$

kde D_t je dělitel (z angl. divisor) a p_t^i je cena akcie v čase t společnosti i .

- Společnosti zahrnuté v DJIA :

```

spolecnostizKR = FinancialData["^DJI", "Members"][[All]];
sektorpodnikani = FinancialData[#, "Sector"] & /@spolecnostizKR;
(spolecnostiAll = Transpose[{FinancialData[#, "Company"] & /@spolecnostizKR,
    spolecnostizKR, sektorpodnikani}]) // TableForm

```

American Express Company	AXP	CreditServices
Boeing Company	BA	Aerospace/Defense-MajorDiversified
Caterpillar Inc.	CAT	FarmAndConstructionMachinery
Cisco Systems, Inc.	CSCO	NetworkingAndCommunicationDevices
Chevron Corporation	CVX	MajorIntegratedOilAndGas
E.I. du Pont de Nemours & Company	DD	Chemicals-MajorDiversified
Walt Disney Company	DIS	Entertainment-Diversified
General Electric Company	GE	Conglomerates
Goldman Sachs Group, Inc.	GS	DiversifiedInvestments
Home Depot, Inc.	HD	HomeImprovementStores
International Business Machines Corp	IBM	DiversifiedComputerSystems
Intel Corporation	INTC	Semiconductor-BroadLine
Johnson & Johnson	JNJ	DrugManufacturers-Major
J.P. Morgan Chase & Co.	JPM	MoneyCenterBanks
Coca-Cola Company	KO	Beverages-SoftDrinks
McDonald's Corporation	MCD	Restaurants
3M Company	MMM	Conglomerates
Merck & Co Inc	MRK	DrugManufacturers-Major
Microsoft Corporation	MSFT	ApplicationSoftware
Nike, Inc.	NKE	Textile-ApparelFootwearAndAccessories
Pfizer Inc.	PFE	DrugManufacturers-Major
Procter & Gamble Company	PG	PersonalProducts
AT&T, Inc.	T	TelecomServices-Domestic
The Travelers Companies, Inc.	TRV	PropertyAndCasualtyInsurance
UnitedHealth Group, Inc.	UNH	HealthCarePlans
United Technologies	UTX	Conglomerates
Visa, Inc.	V	BusinessServices
Verizon Communications Inc.	VZ	TelecomServices-Domestic
Wal-Mart Stores, Inc.	WMT	Discount,VarietyStores
ExxonMobil Corporation	XOM	MajorIntegratedOilAndGas

Shluková analýza - DJIA

- Uvažujme, že investor nechce investovat veškerý svůj kapitál do DJIA, ani do každé společnosti zvlášť, ale chce ho rozdělit mezi některé společnosti, které DJIA zahrnuje.
- U každé společnosti má investor k dispozici 10 finančních ukazatelů (viz níže) ke dni 31.12.2012.
- Na základě těchto ukazatelů chce investor sloučit "nejpodobnější" společnosti do několika shluků a pak se rozhodnout, do kterých shluků a jakým způsobem investovat.
- Pokud jsme se ještě nerozhodli, k určení počtu shluků nám pomůže modifikace známého grafu (scree plot)
- Využijeme knihovnu RLink, která umožňuje spouštět příkazy i funkce z programu R

```

Clear[cesta]
cesta = StringJoin[NotebookDirectory[], "MSA_R.csv"];

RSet["cesta", cesta] (*nastavení cesty k datům*)
C:\Users\Jakub\Documents\MFF\2014_ROBUST\MSA_R.csv

SetDirectory[NotebookDirectory[]];
ukazatele =
  {"Close", "Change", "Volume", "Volatility20Day", "BookValuePerShare",
   "ForwardPERatio", "PriceToBookRatio", "MarketCap", "EarningsPerShare"};
datavypocet = Import["MSA.csv"];
datavypocetstand = Standardize[datavypocet] // N;

REvaluate["{
  data <- read.table(cesta, header = T, sep = "\", \"\", row.names=1)
  soucetctvercu <- 0
  library(cluster)
  set.seed(11)
  for (i in 1:15) soucetctvercu[i]
    <- sum(kmeans(data, centers=i, nstart=30)$withinss)
  plot(1:15, soucetctvercu, type="b", xlab="Počet shluků",
       ylab="Soucet ctvercu")
}"];

(*Propojení s programem R lze provést i pomocí RFunction*)
scree = RFunction["function(shluky) {
  data <- read.table(cesta, header = T, sep = "\", \"\", row.names=1)
  soucetctvercu <- 0
  library(cluster)
  set.seed(11)
  for (i in 1:shluky) soucetctvercu[i]
    <- sum(kmeans(data, centers=i, nstart=30)$withinss)
  plot(1:shluky, soucetctvercu, type="b", xlab="Počet shluků",
       ylab="Soucet ctvercu")
}"];

scree[5]

```

```
scree[20]
```

- počet shluků volme rovný 5

```
pocetshluku = 5;
```

- Rozdělení jednotlivých společností do shluků pomocí funkce **ClusteringComponents** vypadá následovně :

```
shluky = ClusteringComponents[datavypocetstand, pocetshluku, 1,
  Method -> "KMeans", DistanceFunction -> EuclideanDistance, "RandomSeed" -> 16]
shlukydata = Table[Prepend[datavypocet[[i]], shluky[[i]]],
  {i, 1, Length[datavypocet]}];
Sort[Table[Prepend[spolecnostiAll[[i]], shluky[[i]]],
  {i, 1, Length[spolecnostiAll]}, #1[[1]] < #2[[1]] &] // TableForm
{1, 2, 3, 1, 3, 2, 4, 2, 2, 1, 5, 1, 3, 2, 5, 4, 5, 3, 3, 2, 2, 2, 5, 5, 4, 2, 5, 5, 5, 4}

1 Intel Corporation INTC Semiconductor-BroadLine
1 Home Depot, Inc. HD HomeImprovementStores
1 Cisco Systems, Inc. CSCO NetworkingAndCommunicationDe
1 American Express Company AXP CreditServices
2 United Technologies UTX Conglomerates
2 Procter & Gamble Company PG PersonalProducts
2 Pfizer Inc. PFE DrugManufacturers-Major
2 Nike, Inc. NKE Textile-ApparelFootwearAndAc
2 J.P. Morgan Chase & Co. JPM MoneyCenterBanks
2 Goldman Sachs Group, Inc. GS DiversifiedInvestments
2 General Electric Company GE Conglomerates
2 E.I. du Pont de Nemours & Company DD Chemicals-MajorDiversified
2 Boeing Company BA Aerospace/Defense-MajorDiver
3 Microsoft Corporation MSFT ApplicationSoftware
3 Merck & Co Inc MRK DrugManufacturers-Major
3 Johnson & Johnson JNJ DrugManufacturers-Major
3 Chevron Corporation CVX MajorIntegratedOilAndGas
3 Caterpillar Inc. CAT FarmAndConstructionMachinery
4 ExxonMobil Corporation XOM MajorIntegratedOilAndGas
4 UnitedHealth Group, Inc. UNH HealthCarePlans
4 McDonald's Corporation MCD Restaurants
4 Walt Disney Company DIS Entertainment-Diversified
5 Wal-Mart Stores, Inc. WMT Discount,VarietyStores
5 Verizon Communications Inc. VZ TelecomServices-Domestic
5 Visa, Inc. V BusinessServices
5 The Travelers Companies, Inc. TRV PropertyAndCasualtyInsurance
5 AT&T, Inc. T TelecomServices-Domestic
5 3M Company MMM Conglomerates
5 Coca-Cola Company KO Beverages-SoftDrinks
5 International Business Machines Corp IBM DiversifiedComputerSystems
```

- Pro nehierarchické shlukování můžeme také použít funkci **FindCluster**, pro hierarchické shlukování pak existuje balík **Needs["HierarchicalClustering`"]**.

Výsledky

- Průměrné hodnoty ve slucích

```
TableForm[Transpose[Table[
  Mean[Map[Rest, Select[shlukydata, #[[1]] == i &]], {i, 1, pocetshluku}]],
  TableHeadings → {Flatten[{ukazatele, "Return"}], Range[pocetshluku]}] // N;
```

- Na grafu prvních dvou hlavních komponent si zobrazíme shluky, do kterých jsme společnosti DJIA rozdělili

```
pocetshluku;

RSet["pocetshluku", pocetshluku];

ToRForm[{spolecnostizKR, shluky}]

RList[
  {RVector[character, {AXP, BA, CAT, CSCO, CVX, DD, DIS, GE, GS, HD, IBM, INTC, JNJ,
    JPM, KO, MCD, MMM, MRK, MSFT, NKE, PFE, PG, T, TRV, UNH, UTX, V, VZ, WMT, XOM},
    RAttributes[]], RVector[integer, {1, 2, 3, 1, 3, 2, 4, 2, 2, 1, 5, 1, 3, 2,
    5, 4, 5, 3, 3, 2, 2, 2, 5, 5, 4, 2, 5, 5, 5, 4}], RAttributes[]}, RAttributes[]]

shlukyR = RObject[shluky,
  RAttributes["names" => {"AA", "AXP", "BA", "BAC", "CAT", "CSCO", "CVX", "DD",
    "DIS", "GE", "HD", "HPQ", "IBM", "INTC", "JNJ", "JPM", "KO", "MCD", "MMM",
    "MRK", "MSFT", "PFE", "PG", "T", "TRV", "UNH", "UTX", "VZ", "WMT", "XOM"}]];

RSet["shlukyR", shlukyR];

REvaluate["{
  library(cluster)
  clusplot(data, shluky, color=TRUE,
    shade=TRUE, labels=2, lines=0, main = \"Cluster Plot\")
}"];

(*Shluková analýza v Rku*)
REvaluate["{
shlukovanalyza<-kmeans(data,5)
library(cluster)
clusplot(data,shlukovanalyza$cluster,color=TRUE,shade=TRUE,labels=2,lines=0,
  main = \"Cluster Plot\")}"];

```

Mathematica 9 a SQL (DatabaseLink)

- DatabaseLink zprostředkovává spojení mezi Mathematicou a databázemi
- Obsahuje “malou” databázi, pokud uživatel ještě nemá k dispozici jinou
- Podporuje mnoho databází (databázových systémů) : MySQL, PostgreSQL, SQL Server, Oracle, Sybase a další
- Nabízí uživatelské rozhraní pro generování SQL příkazů v Mathematice, tzv. Database Explorer
- Podpora hromadných (batch) operací při práci s velkými daty
- Pracuje se standardními databázovými typy : čísla, řetězce, binary, datetime,...

DatabaseLink základy

- Nejdříve je třeba nastavení připojení pomocí balíku DatabaseLink

```
Needs["DatabaseLink`"];
```

- Před tím, než si ukážeme Mathematické funkce, si ukážeme uživatelské rozhraní, ve kterém se dají některé příkazy "naklikat" a následně vložit do notebooku.

```
DatabaseExplorer[]
```

- Funkce **OpenSQLConnection** otevře připojení do zadané databáze. Konkrétně se připojíme a budeme pracovat s databází v Mathematice, která se jmenuje demo.

```
conn = OpenSQLConnection["demo"]
```

```
SQLConnection[conn, 1, Open, TransactionIsolationLevel -> ReadCommitted]
```

- Vytvoření tabulky TestTabulka v databázi

```
SQLCreateTable[conn, "TestTabulka",
  {SQLColumn["Cislo", "DataTypeName" -> "INTEGER", Nullable -> True],
   SQLColumn["Jmeno", "DataTypeName" -> "CHAR"],
   SQLColumn["Prijem", "DataTypeName" -> "FLOAT"]}];
```

- Vložení dat do tabulky - funkce **SQLInsert**

```
Clear[pocet, datatabulka];
```

```
pocet = 10;
```

```
datatabulka =
```

```
{Range[pocet], Map[StringJoin[{"Jméno"}, ToString[#]] &, Range[pocet]],
 RandomReal[{10 000, 100 000}, pocet]} // Transpose
```

```
{1, Jméno1, 99 884.5}, {2, Jméno2, 60 535.3},
 {3, Jméno3, 64 192.6}, {4, Jméno4, 88 903.5}, {5, Jméno5, 84 396.3},
 {6, Jméno6, 47 617.3}, {7, Jméno7, 73 389.3}, {8, Jméno8, 32 521.1},
 {9, Jméno9, 55 706.}, {10, Jméno10, 40 748.7}}
```

```
SQLInsert[conn, "TestTabulka", {"Cislo", "Jmeno", "Prijem"}, datatabulka]
```

```
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
```

- Výběr dat z tabulky - funkce **SQLSelect**

```
SQLSelect[conn, "TestTabulka", "ShowColumnHeadings" -> True] // TableForm
```

CISLO	JMENO	PRIJEM
1	Jméno1	99 884.5
2	Jméno2	60 535.3
3	Jméno3	64 192.6
4	Jméno4	88 903.5
5	Jméno5	84 396.3
6	Jméno6	47 617.3
7	Jméno7	73 389.3
8	Jméno8	32 521.1
9	Jméno9	55 706.
10	Jméno10	40 748.7

- Přidání dalších dat do tabulky

```

Clear[cisla, jmena];
cisla = Range[pocet, pocet + 10, 1];
jmena = Map[StringJoin[{"Jméno"}, ToString[#]] &, cisla];
prijem = RandomReal[{10 000, 100 000}, Length[cisla]];

novadata = Transpose[{cisla, jmena, prijem}];

SQLInsert[conn, "TestTabulka", {"Cislo", "Jmeno", "Prijem"}, novadata];

SQLSelect[conn, "TestTabulka", "ShowColumnHeadings" → True] // TableForm

```

CISLO	JMENO	PRIJEM
1	Jméno1	99 884.5
2	Jméno2	60 535.3
3	Jméno3	64 192.6
4	Jméno4	88 903.5
5	Jméno5	84 396.3
6	Jméno6	47 617.3
7	Jméno7	73 389.3
8	Jméno8	32 521.1
9	Jméno9	55 706.
10	Jméno10	40 748.7
10	Jméno10	27 629.8
11	Jméno11	62 492.7
12	Jméno12	37 564.7
13	Jméno13	35 906.3
14	Jméno14	40 158.1
15	Jméno15	46 459.8
16	Jméno16	60 867.6
17	Jméno17	65 890.2
18	Jméno18	63 777.3
19	Jméno19	50 268.5
20	Jméno20	74 065.2

■ Udatování dat - funkce **SQLUpdate**

```

SQLUpdate[conn, "TestTabulka", {"Cislo", "Jmeno", "Prijem"},
  {999, "Jméno", 0}, SQLColumn["Prijem"] > 50 000];

```

```
SQLSelect[conn, "TestTabulka", "ShowColumnHeadings" → True] // TableForm
```

CISLO	JMENO	PRIJEM
6	Jméno6	47 617.3
8	Jméno8	32 521.1
10	Jméno10	40 748.7
10	Jméno10	27 629.8
12	Jméno12	37 564.7
13	Jméno13	35 906.3
14	Jméno14	40 158.1
15	Jméno15	46 459.8
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.
999	Jméno	0.

■ Mazání dat v tabulce

```
SQLDelete[conn, "TestTabulka", SQLColumn["Cislo"] == 999]
```

```
13
```

```
SQLSelect[conn, "TestTabulka", "ShowColumnHeadings" → True] // TableForm
```

CISLO	JMENO	PRIJEM
6	Jméno6	47 617.3
8	Jméno8	32 521.1
10	Jméno10	40 748.7
10	Jméno10	27 629.8
12	Jméno12	37 564.7
13	Jméno13	35 906.3
14	Jméno14	40 158.1
15	Jméno15	46 459.8

■ Mazání tabulky

```
SQLDropTable[conn, "TestTabulka"];
```

■ Využití SQL skriptů

```
connpublish = OpenSQLConnection["publisher"];
```

```
SQLExecute[connpublish, "Select * from TITLES",
  "ShowColumnHeadings" → True] // TableForm
```

TITLE_ID	TITLE	TYPE	P
BS1001	Designer Class Action Suits	business	0
PY2002	Self Hypnosis: A Beginner's Guide	psychology	0
PY2003	Phobic Psychology	psychology	0
UK3004	Hamburger Again!	mod_cook	0
CK4005	Made to Wonder: Cooking the Macabre	trad_cook	0
UK3006	How to Burn a Compact Disk	Null	0
BS1007	Modems for Morons	business	1
PY2008	How Green Is My Valley?	psychology	0
CP5009	The Net: Feeding Trolls and Eating Spam	popular_comp	1
CP5010	Taiwan Trails	popular_comp	1
BS1011	Guide to Impractical Databases	business	1
PY2012	Know Thyself	psychology	0
PY2013	Where Minds Meet: The Impact of Diet on Behavior	psychology	0
BS1014	Exit Interviews	business	1
UK3015	Treasures of the Sierra Madre	mod_cook	0
CK4016	Too Many Cooks	trad_cook	0
CK4017	Let Them Eat Cake!	trad_cook	0
CP5018	Sticky Software: UI and GUI	popular_comp	1

```
SQLExecute[connpublish,
```

```
  "Select PUB_ID, SUM(PRICE) as TotalPrice, COUNT(*) as Cnt from
  TITLES group by PUB_ID", "ShowColumnHeadings" → True] // TableForm
```

PUB_ID	TOTALPRICE	CNT
0736	70.51	6
0877	70.87	6
1389	94.88	6

DatabaseLink - příklad

- Otevření externí databáze pomocí JDBC

```
Needs["DatabaseLink`"];
(*Needs["JLink`"];
AddToClassPath["C:\\Program Files\\Microsoft JDBC Driver
  4.0 for SQL Server\\sqljdbc_4.0\\enu\\sqljdbc4.jar"];*)
sqlcon = OpenSQLConnection[JDBC["net.sourceforge.jtds.jdbc.Driver",
  "jdbc:jtds:sqlserver://JAKUB-NTB:1433/AdventureWorks2012;instance=
  SQLEXPRESS"], "Username" -> "sa", "Password" -> "qwertz123"]
SQLConnection[1, Open, Catalog -> AdventureWorks2012,
  TransactionIsolationLevel -> ReadCommitted]
```

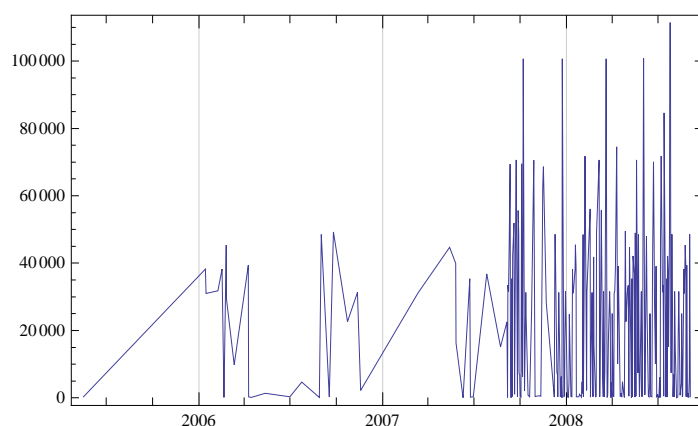
Načtení časové řady z dostupné databáze AdventureWorks2012 a její zobrazení

- Uložení časové řady do proměnné query

```
Clear[query];
query =
  SQLExecute[sqlcon, "SELECT EmployeeID,OrderDate,sum(TotalDue) as Total
FROM[AdventureWorks2012].[Purchasing].[PurchaseOrderHeader]
group by EmployeeID,OrderDate
having EmployeeID=254 order by OrderDate"] /. SQLDateTime -> Identity;
```

- Jedná se o zisk připadající na jednoho zaměstnance za určité období, podívejme se vývoj v čase

```
Clear[dataprodej];
dataprodej = ToExpression[Drop[#, 1] & /@ query];
DateListPlot[dataprodej, Joined -> True]
```



Hledání adresy dle zeměpisné šířky a délky

- Uložení zeměpisné šířky a délky do proměnné query1

```
Clear[query1];
query1 =
  SQLExecute[sqlcon, "Select distinct top 5000 SpatialLocation.Lat as Lat,
    SpatialLocation.Long as Long
  from Person.Address a INNER JOIN Sales.SalesOrderHeader b
  on a.AddressID=b.ShipToAddressID"];

```

■ Zobrazení adres na mapě

```
Graphics[{Red, Point /@
  Map[GeoGridPosition[GeoPosition[#, "Mercator"][[1]] &, {query1}, {2}],
  Gray, Polygon[Map[GeoGridPosition[GeoPosition[#, "Mercator"][[1]] &,
    CountryData[#, "Coordinates"], {2}]] & /@ CountryData["Countries"]]}]

```



Doplnění jmen měn a aktuálních kurzů

```
Clear[currID, seznammen, menyajmena, kurzy];
(currID = SQLExecute[sqlcon,
  "Select CurrencyCode, Name, ModifiedDate from [Sales].[Currency]",
  "ShowColumnHeadings" → True]) // TableForm;
seznammen = Intersection[First[Transpose[currID]],
  FinancialData["Currencies"]];
kurzy = FinancialData[StringJoin[{"#", "/CZK"}]] & /@ seznammen;

menyajmena =
  Flatten[SQLSelect[sqlcon, "[Sales].[Currency]", {"CurrencyCode", "Name"},
    SQLColumn["CurrencyCode"] == #] & /@ seznammen, 1];

```

```
TableForm[Transpose[Join[Transpose[menyajmena], {kurzy},
  {ConstantArray[DateString[DateString[{"DayNameShort", " ", "Day",
    " ", "MonthNameShort", " ", "Year"}]], Length[menyajmena]}]],
  TableHeadings → {None, {currID[[1, 1]], currID[[1, 2]],
    "CurrencyRate", currID[[1, 3]]}}]
```

CurrencyCode	Name	CurrencyRate	Modifie
AED	Emirati Dirham	5.5082	Thu 16
ALL	Lek	0.1953	Thu 16
AMD	Armenian Dram	0.0498	Thu 16
ANG	Netherlands Antillian Guilder	11.3028	Thu 16
AOA	Kwanza	0.2072	Thu 16
ARS	Argentine Peso	2.987	Thu 16
AUD	Australian Dollar	17.8159	Thu 16
AWG	Aruban Guilder	11.2996	Thu 16
BBD	Barbados Dollar	10.116	Thu 16
BDT	Taka	0.2617	Thu 16
BGN	Bulgarian Lev	14.0559	Thu 16
BHD	Bahraini Dinar	53.6558	Thu 16
BND	Brunei Dollar	15.9	Thu 16
BOB	Boliviano	2.9279	Thu 16
BRL	Brazilian Real	8.5511	Thu 16
BSD	Bahamian Dollar	20.232	Thu 16
BTN	Ngultrum	0.3285	Thu 16
CAD	Canadian Dollar	18.5153	Thu 16
CHF	Swiss Franc	22.2888	Thu 16
CLP	Chilean Peso	0.0379	Thu 16
CNY	Yuan Renminbi	3.3401	Thu 16
COP	Colombian Peso	0.0104	Thu 16
CRC	Costa Rican Colon	0.0405	Thu 16
CZK	Czech Koruna	1	Thu 16
DKK	Danish Krone	3.6833	Thu 16
DOP	Dominican Peso	0.4726	Thu 16
DZD	Algerian Dinar	0.2579	Thu 16
EEK	Kroon	Missing[NotAvailable]	Thu 16
EGP	Egyptian Pound	2.9049	Thu 16
EUR	EURO	27.484	Thu 16
FJD	Fiji Dollar	10.8073	Thu 16
GBP	United Kingdom Pound	33.0212	Thu 16
GTQ	Quetzal	2.5824	Thu 16
HKD	Hong Kong Dollar	2.6083	Thu 16
HRK	Croatian Kuna	3.6038	Thu 16
HUF	Forint	0.0914	Thu 16
IDR	Rupiah	0.0017	Thu 16
ILS	New Israeli Shekel	5.7938	Thu 16
INR	Indian Rupee	0.3287	Thu 16
ISK	Iceland Krona	0.1752	Thu 16
JMD	Jamaican Dollar	0.1904	Thu 16
JOD	Jordanian Dinar	28.5996	Thu 16
JPY	Yen	0.1939	Thu 16
KES	Kenyan Shilling	0.2359	Thu 16
KRW	Won	0.019	Thu 16
KWD	Kuwaiti Dinar	71.5311	Thu 16
LBP	Lebanese Pound	0.0134	Thu 16
LKR	Sri Lankan Rupee	0.1548	Thu 16
LTL	Lithuanian Litas	7.9614	Thu 16
LVL	Latvian Lats	39.6201	Thu 16
MAD	Moroccan Dirham	2.4517	Thu 16
MUR	Mauritius Rupee	0.6708	Thu 16

MVR	Rufiyaa	1.3146	Thu 16
MXN	Mexican Peso	1.521	Thu 16
MYR	Malaysian Ringgit	6.1405	Thu 16
NAD	Namibia Dollar	1.8643	Thu 16
NGN	Naira	0.1265	Thu 16
NOK	Norwegian Krone	3.2633	Thu 16
NPR	Nepalese Rupee	0.2054	Thu 16
NZD	New Zealand Dollar	16.847	Thu 16
OMR	Omani Rial	52.5435	Thu 16
PAB	Balboa	20.232	Thu 16
PEN	Nuevo Sol	7.2	Thu 16
PHP	Philippine Peso	0.4482	Thu 16
PKR	Pakistan Rupee	0.1919	Thu 16
PLN	Zloty	6.5918	Thu 16
PYG	Guarani	0.0044	Thu 16
RUB	Russian Ruble	0.6052	Thu 16
SAR	Saudi Riyal	5.3935	Thu 16
SEK	Swedish Krona	3.1238	Thu 16
SGD	Singapore Dollar	15.8991	Thu 16
SKK	Slovak Koruna	Missing [NotAvailable]	Thu 16
THB	Baht	0.6173	Thu 16
TND	Tunisian Dinar	12.2278	Thu 16
TTD	Trinidad and Tobago Dollar	3.1559	Thu 16
TWD	New Taiwan Dollar	0.6721	Thu 16
USD	US Dollar	20.229	Thu 16
UYU	Uruguayan Peso	0.9493	Thu 16
VND	Dong	0.001	Thu 16
XOF	CFA Franc BCEAO	0.042	Thu 16
ZAR	Rand	1.85	Thu 16
ZWD	Zimbabwe Dollar	Missing [NotAvailable]	Thu 16

Generování procesů

Obecné difúzní procesy

- V nové verzi Mathematicy lze také generovat obecně definované procesy.

Itoův proces

- Uvažujeme Itoův proces $(X(t), t \geq 0)$ ve standardním tvaru, tj. $dX(t) = a(t, X(t)) dt + b(t, X(t)) dW(t)$, nebo v integrálním tvaru $X(t) = X(t_0) + \int_{t_0}^t a(u, x(u)) du + \int_{t_0}^t b(u, x(u)) dW(u)$
- Pokud je drift a n -rozměrný vektor a difúzní koeficient b matice o rozměrech $n \times m$, proces je n -rozměrný a je řízen m -rozměrným Brownovým pohybem .
- Defaultně je čas t_0 nastaven na nulu a kovarianční matice u Brownova pohybu je jednotková.

Itoův proces - příklad

- Jednorozměrný Itoův proces $dX(t) = -(X(t) - \frac{1}{2}) dt + \sqrt{1 + X(t)^2} dW(t)$, kde $X(t_0) = 1$ a $(W(t), t \geq 0)$ je Wienerův proces

```
Clear[itoproc];
```

```
Clear[x];
```

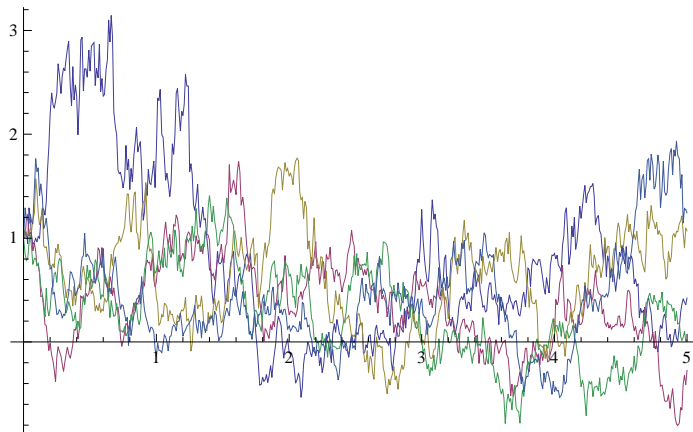
```
itoproc = ItoProcess[dx[t] == -(x[t] - 1/2) dt + Sqrt[1 + x[t]^2] dW[t],  
  x[t], {x, 1}, t, W ~ WienerProcess[]];
```

- 5 simulací procesu od 0 do 5 s krokem 0.01

```
pocetsimulaci = 5;
```

```
Table[RandomFunction[itoproc, {0., 5., 0.01}], {i, 1, 5}];
```

```
ListLinePlot[%]
```



- Výpočet střední hodnoty procesu v čase t

```
Mean[itoproc[t]]
```

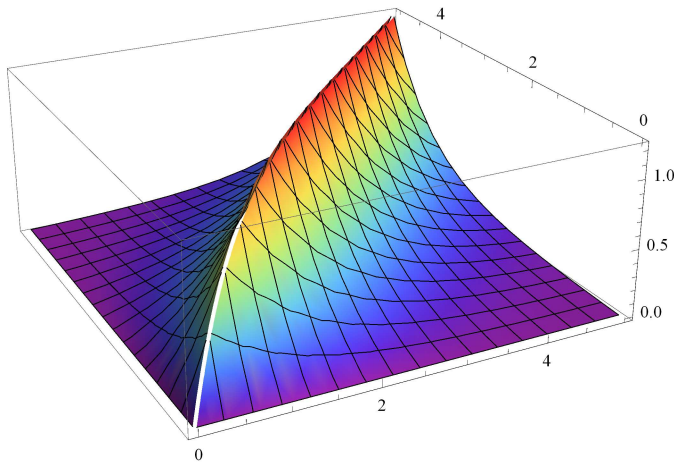
$$\frac{1}{2} (1 + e^{-t})$$

- Výpočet autokovarianční funkce procesu

```
CovarianceFunction[itoproc, s, t]
```

$$\frac{1}{4} e^{-s-t} (-1 - 4 e^{\text{Min}[s,t]} + 5 e^{2\text{Min}[s,t]} + 2 e^{\text{Min}[s,t]} \text{Min}[s, t])$$

```
Plot3D[%, {s, 0, 5}, {t, 0, 5}, ColorFunction -> "Rainbow"]
```



Stratonovichův proces

- Uvažujeme Stratonovichův proces $(X(t), t \geq 0)$ ve standardním tvaru, tj. $dX(t) = a(t, X(t)) dt + b(t, X(t)) \circ dW(t)$,
- Pokud je drift a n -rozměrný vektor a difúzní koeficient b matice o rozměrech $n \times m$, proces je n -rozměrný a je řízen m -rozměrným Brownovým pohybem .
- Defaultně je čas t_0 nastaven na nulu a kovarianční matice u Brownova pohybu je jednotková.

Stratonovichův proces - příklad

- Pro porovnání nastavíme koeficienty Stratonovichova procesu stejně, jako u Itoova.

```
Clear[stratproc];
```

```
stratproc = StratonovichProcess[dx[t] == -(x[t] - 1/2) dt + Sqrt[1 + x[t]^2] dW[t],  
x[t], {x, 1}, t, W ~ WienerProcess[]]
```

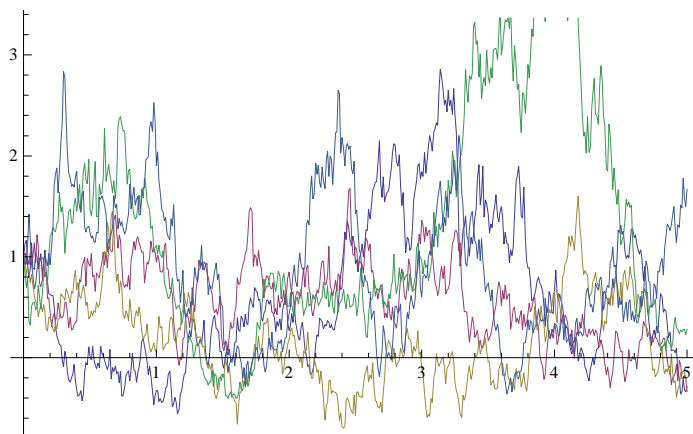
```
StratonovichProcess[{{1/2 (1 - 2 x[t])}}, {{Sqrt[1 + x[t]^2]}}, x[t], {{x}, {1}}, {t, 0}]
```

- 5 simulací procesu od 0 do 5 s krokem 0.01

```
pocetsimulaci = 5;
```

```
Table[RandomFunction[stratproc, {0., 5., 0.01}], {i, 1, pocetsimulaci}];
```

```
ListLinePlot[%]
```



- Výpočet střední hodnoty procesu v čase t

```
Mean[stratproc[t]]
```

```
1
```

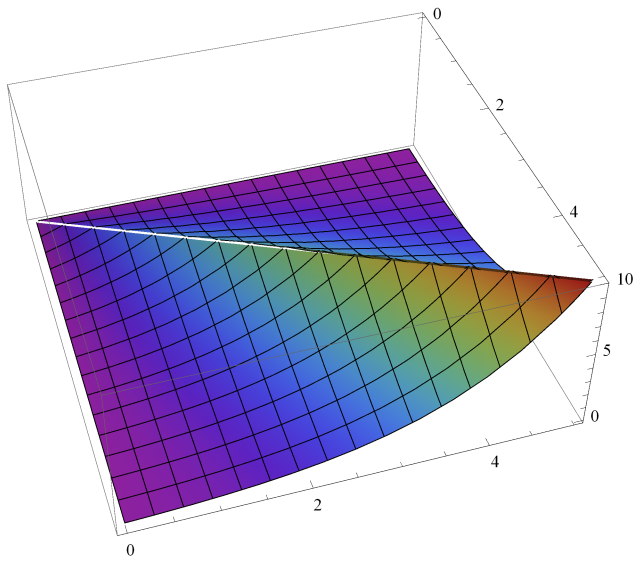
- Výpočet a zobrazení autokovarianční funkce procesu

```
CovarianceFunction[stratproc, s, t]
```

```
 $2 e^{-\frac{s-t}{2} + \text{Min}[s, t]} \text{Min}[s, t]$ 
```



```
Plot3D[%, {s, 0, 5}, {t, 0, 5}, ColorFunction -> "Rainbow"]
```



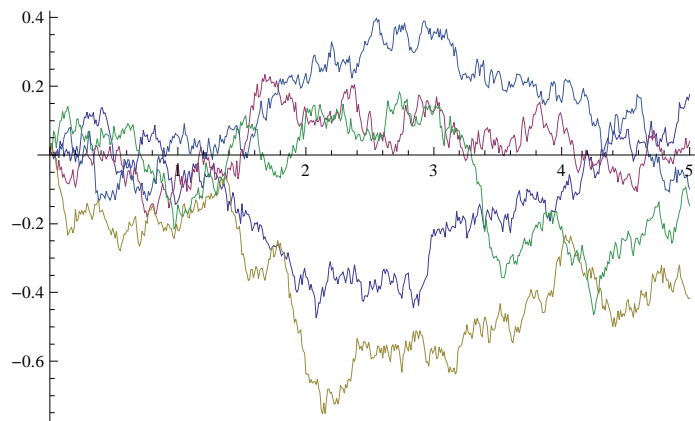
Konkrétní difúzní procesy

- Zběžně si ukážeme i některé konkrétní procesy.

Ornstein-Uhlenbeck proces

- Jedná se o Ornstein-Uhlenbeck proces s konstantními koeficienty (ve financích známý jako Vašíčkův model pro vývoj úrokových sazeb), tj. $(X(t), t \geq 0)$ je Ornstein-Uhlenbeck proces, pokud se řídí SDR ve tvaru $dX(t) = \theta(\mu - X(t))dt + \sigma dW(t)$, kde W je Wienerův proces, μ je dlouhodobý (návrátový průměr), θ je rychlost návratu a σ je difúzní koeficient.

```
Clear[oup];
oup = Table[RandomFunction[OrnsteinUhlenbeckProcess[0, .2, .4, 0.01],
  {0, 5, .01}], {i, 1, pocetsimulaci}];
ListLinePlot[%]
```



- Výpočet střední hodnoty procesu v čase t se startovní hodnotou $X(t_0) = X_0$

```
Mean[OrnsteinUhlenbeckProcess[μ, σ, θ, x₀][t]]
```

$$(1 - e^{-t\theta}) \mu + e^{-t\theta} X_0$$

- Výpočet rozptylu procesu v čase t se startovní hodnotou $X(t_0) = X_0$

```
Variance[OrnsteinUhlenbeckProcess[μ, σ, θ, x₀][t]]
```

$$\frac{(1 - e^{-2t\theta}) \sigma^2}{2\theta}$$

- Určení rozdělení tohoto procesu v čase t se startovní hodnotou $X(t_0) = X_0$

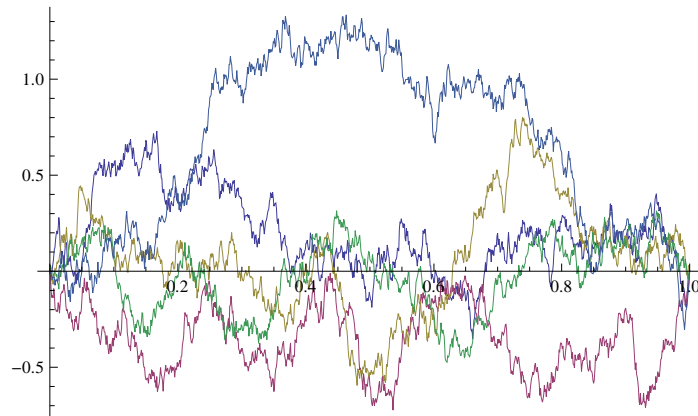
```
sliceDistribution[OrnsteinUhlenbeckProcess[μ, σ, θ, x₀], t]
```

$$\text{NormalDistribution}\left[\left(1 - e^{-t\theta}\right) \mu + e^{-t\theta} x_0, \frac{\sqrt{\frac{(1 - e^{-2t\theta}) \sigma^2}{\theta}}}{\sqrt{2}}\right]$$

Brownův most

- V případě, že funkce nemá žádné argumenty, jedná se o klasický Brownův most definovaný jako $B(t) = W(t) - tW(1)$, kde $(W(t), t \geq 0)$, je Wienerův proces.

```
Clear[bbp];
bbp = Table[RandomFunction[BrownianBridgeProcess[], {0, 1, 0.001}],
  {i, 1, pocetsimulaci}];
ListLinePlot[
  %]
```



- Výpočet střední hodnoty procesu v čase t se startovní hodnotou $X(t_0) = X_0$

```
Mean[BrownianBridgeProcess[]][t]
```

0

- Výpočet rozptylu procesu v čase t se startovní hodnotou $X(t_0) = X_0$

```
Variance[BrownianBridgeProcess[]][t]
```

$-(-1+t)t$

```
CovarianceFunction[BrownianBridgeProcess[], s, t]
```

$-st + \text{Min}[s, t]$

- Pokud přidáme argumenty, získáme tím obecnější tvar Brownova mostu, tj. Brownův most s volatilitou σ na intervalu (t_1, t_2) , kde $B(t_1)=a$ a $B(t_2)=b$, potom charakterisky vypadají následovně

```
Mean[BrownianBridgeProcess[\sigma, {t1, a}, {t2, b}][t]
```

$a + \frac{(-a+b)(t-t_1)}{-t_1+t_2}$

```
Variance[BrownianBridgeProcess[\sigma, {t1, a}, {t2, b}][t]
```

$((t-t_1)(t-t_2)\sigma^2) / (t_1-t_2)$

```
SliceDistribution[BrownianBridgeProcess[\sigma, {t1, a}, {t2, b}], t]
```

$\text{NormalDistribution}[a + ((-a+b)(t-t_1)) / (-t_1+t_2),$
 $\sqrt{(((t-t_1)(t-t_2)) / (t_1-t_2)) \sigma}]$

Zdroje

- DatabaseLink - DatabaseLink User Guide
- RLink - RLink User Guide
- <http://coldlogics.wordpress.com>
- <http://www.wolfram.com/>

Děkuji za pozornost.

Offline data