Statistika v PostgreSQL pomocí R-project

Vratislav Beneš
benes@optisolutions.cz

1. **Potřeba statistiky v praxi**
2. **Proč řešit statistické analýzy v DB serveru**
3. **Možnosti SQL v PostgreSQL**
4. **R-project**
5. **Propojení R-project s PostgreSQL**
6. **Jazyk plr**
7. **Příklad analýzy predikce v krátkém období**
8. ***-- SQL dotazy použité při prezentaci --***

Potřeba statistiky v praxi

PREDIKTIVNÍ ANALYTIKA

Obchodní firmy (CRM)

- Odhad vývoje příležitostí
- Analýza chování zákazníka

Výrobní firmy (ERP)

- Plánovaní výroby
- Analýzy důvodů poruchovosti linek
- Predikce poruchovosti linek

Obchodní oddělení

- Odhad vývoje kurzu měn

Marketingová oddělení

- Analýzy a modelování nad daty z průzkumů trhů

Proč řešit statistické analýzy v DB serveru

- **Snadná dostupnost z aplikací**
- **Žádné speciální nároky na aplikaci/aplikační server**
- **Výkonnost**
- **Škálovatelnost**
- **Rozšiřitelnost**
- **Snadná údržba**
- **SQL – častokrát plně postačí**

Možnosti SQL v PostgreSQL

- **Agregační funkce**
  - AVG, směrodatné odchylky
  - Lineární regrese
    - Jednouché predikce
    - Trendy
  - Korelace
- **Window funkce – od verze 8.4**
  - Klouzavé průměry
  - Odpadají nadbytečné joiny
  - Zjednodušení práce

R-project

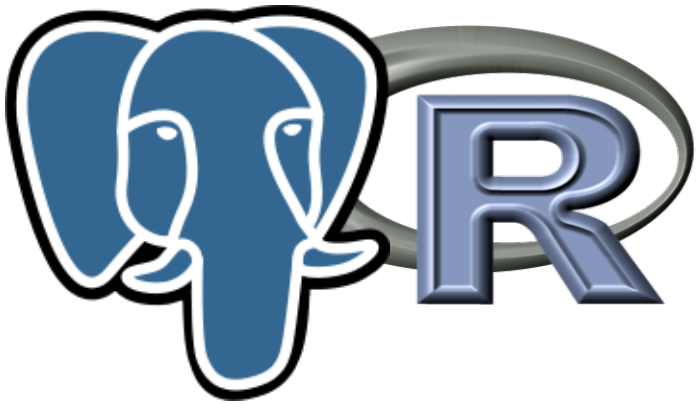
- **Opensource projekt pro statistiku pod GPL - www.r-project.org**
- **R–jazyk implementující S–jazyk**
- **Modulární systém**
  - Pomocí balíčků - cran.r-project.org/web/packages
  - Časové řady
  - Bayesovské sítě
  - Neuronové sítě
  - Nelineární modely
  - ...
- **TUI, GUI**
- **Podpora distribuovaných výpočtů**
- **Podpora GPU**
- **Multiplatformní**
- **Silná komunita**

Propojení R-project s PostgreSQL

- Jazyk do PostgreSQL
- **http://www.joeconway.com/plr/**

Instalace

- Nakopírovat do contrib a skompilovat
- Statická knihovna ***plr.so***
- Import ***plr.sql*** do každé DB, kde má být plr používán
- Jednoduchý test: *select r_version();*



Jazyk plr

```
CREATE OR REPLACE FUNCTION r_arima_x(x double precision[], per integer)
  RETURNS double precision[] AS
$BODY$
  myx<-ts(x,start=2008,frequency=12)
  myx.hw<-HoltWinters(myx)
  predict(myx.hw,n.ahead=per)


  $BODY$
  LANGUAGE plr VOLATILE
  COST 100;
ALTER FUNCTION r_arima_x(double precision[], integer) OWNER TO postgres;
```

- Do těla funkce vepsat R kód
- **Pozor na typ návratové hodnoty skalár/vektor**
- Využívat array
  - SELECT array(*SELECT x FROM TBL)* -> array
  - UNNEST array -> recordset
- Funkce s návratovým typem recordset

Příklad analýzy predikce v krátkém období

Zadání:

Vypracovat technickou analýzu pro kráktokobé predikce 500 výrobků

Zdrojová data:

- ~12GB
- ~25M řádků
- 12 atributů

Výstup: dynamický systém reflektující na změnu vstupních dat s výstupy do tabulky a grafu

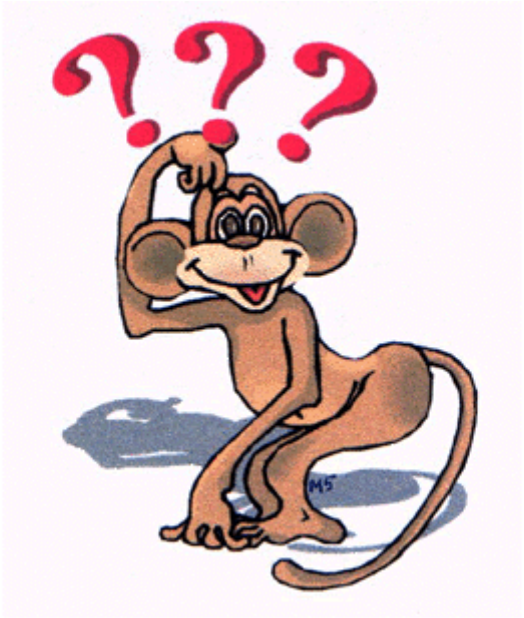
Řešení:

- **PostgreSQL 8.4 a R-project**
- **ARIMA, vícerozměrná lineární regrese, korelace**
- **Doba realizace 3 týdny**
- **Doba výpočtu na Opteron 3.4GHz ~ 10min**
- **Vizualizace pomocí Pentaho**

Příklady

```
--nejnizsi namerene teploty v jednotlivych mesicich
select datum, teplota
From (select rank() over (PARTITION by  extract(year from datum)*100 + extract(month from datum) order by
teplota),* from pocasi) t1
where t1.rank=1

--klouzavy prumer
select yd, t,  avg(t) OVER (order BY yd ROWS BETWEEN 3 PRECEDING AND 1 PRECEDING) t from
(select extract(year from datum)*1000 + extract(doy from datum) yd, avg(teplota) t
from pocasi
group by yd order by yd) a


-- vytvoreni funkce
CREATE OR REPLACE FUNCTION r_arima_x(x double precision[], per integer)
  RETURNS double precision[] AS
$BODY$
  myx<-ts(x,start=1,frequency=48)
  myx.hw<-HoltWinters(myx)
  predict(myx.hw,n.ahead=per)
  $BODY$
  LANGUAGE plr VOLATILE
  COST 100;
ALTER FUNCTION r_arima_x(double precision[], integer) OWNER TO postgres;


--zavolani fce
select r_arima_x(array(select teplota from pocasi),48)

--tabulka vysledku
select unnest(r_arima_x(array(select teplota from pocasi),48))
```

Otázky



Děkuji za pozornost

Vratislav Beneš
benes@optisolutions.cz