

# PostgreSQL-embedded Statistical Analysis with PL/R

**Joe Conway**  
mail@joeconway.com

**PGDay**  
July 19, 2009

# Agenda

- Introduction
- New Features
- Examples

# Introduction

- What is PL/R?
  - R Procedural Language for PostgreSQL. Enables user-defined SQL functions to be written in the R language
- What is R?
  - R is an open source (GPL) language and environment for statistical computing and graphics. It is similar to the S language and environment, which was developed at Bell Laboratories by John Chambers and colleagues, and was distributed by Insightful Corp as S-Plus. R provides a wide variety of statistical (linear and nonlinear modeling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.
  - Off-topic, but R has excellent package system. I think we should study and emulate parts of it.

## Newest Features

- RPostgreSQL compatibility functions
  - cut-n-paste from R console
- Return serialized R objects as bytea
  - persist pre-calculated data
- Convert bytea arguments to original R object
  - perform final analysis
- Unserialize R object in Postgres
  - restore object outside of R
  - useful for image data

# Returning Rows

- PL/R Function

- get Hi-Low-Close data from Yahoo for CYMI
- store in PostgreSQL table

```
CREATE TABLE stock_data
(symbol text, trade_date date, open float, high float,
low float, close float, volume float, adjusted float);
```

```
CREATE FUNCTION get_stock_data_tuples(symbol text)
RETURNS setof stock_data AS $$
library(quantmod)
mysym<-getSymbols(c(symbol))
mydf<-data.frame(mget(mysym, .GlobalEnv))
mydf <- cbind(symbol, rownames(mydf),mydf)
return(mydf)
$$ LANGUAGE plr;
```

```
INSERT INTO stock_data
SELECT * FROM get_stock_data_tuples('CYMI');
```

# RPostgreSQL Compatibility

- Retrieve CYMI stock data

- R Console

```
library(RPostgreSQL)
drv <- dbDriver("PostgreSQL")
con <- dbConnect(drv, user="postgres", dbname="pgday")
df <- dbGetQuery(con, "select * from stock_data")
[...]
```

- Equivalent PL/R function

```
CREATE FUNCTION plot_stock_data() RETURNS bytea AS $$
  library(RPostgreSQL)
  drv <- dbDriver("PostgreSQL")
  con <- dbConnect(drv, user="postgres", dbname="pgday")
  df <- dbGetQuery(con, "select * from stock_data")
  [...]
$$ LANGUAGE plr;
```

# Direct Image Return - quantmod Example

- PL/R Function

- get Hi-Low-Close data from Yahoo for any stock symbol
- plot with Bollinger Bands and volume

```
CREATE FUNCTION plot_stock_data(sym text) RETURNS bytea AS $$  
  library(quantmod)  
  library(cairoDevice)  
  library(RGtk2)  
  
  pixmap <- gdkPixmapNew(w=500, h=500, depth=24)  
  asCairoDevice(pixmap)  
  getSymbols(c(sym))  
  chartSeries(get(sym), name=sym, theme="white",  
              TA="addVo();addBBands();addCCI()")  
  plot_pixbuf <- gdkPixbufGetFromDrawable(NULL, pixmap,  
                                           pixmap$getColormap(),0, 0, 0, 0, 500, 500)  
  buffer <- gdkPixbufSaveToBufferv(plot_pixbuf, "jpeg",  
                                   character(0),character(0))$buffer  
  return(buffer)  
$$ LANGUAGE plr;
```

# Direct Image Return - quantmod Example

- Calling it from PHP for CYMI

```
<?php
$dbconn = pg_connect("...");
$rs = pg_query( $dbconn,
    "select plr_get_raw(plot_stock_data('CYMI'))");
$hexpic = pg_fetch_array($rs);
$cleandata = pg_unescape_bytea($hexpic[0]);

header("Content-Type: image/png");
header("Last-Modified: " .
    date("r", filectime($_SERVER['SCRIPT_FILENAME'])));
header("Content-Length: " . strlen($cleandata));
echo $cleandata;

?>
```



# Direct Image Return - quantmod Example



Questions?