

The background of the slide features a repeating geometric pattern of overlapping triangles. Each triangle contains a solid-colored circle. The triangles and circles are in various shades of gray, creating a complex, tessellated effect.

Administration of RStudio Connect in Production

Jeff Allen 3/2017

RStudio Connect



Publish Document...

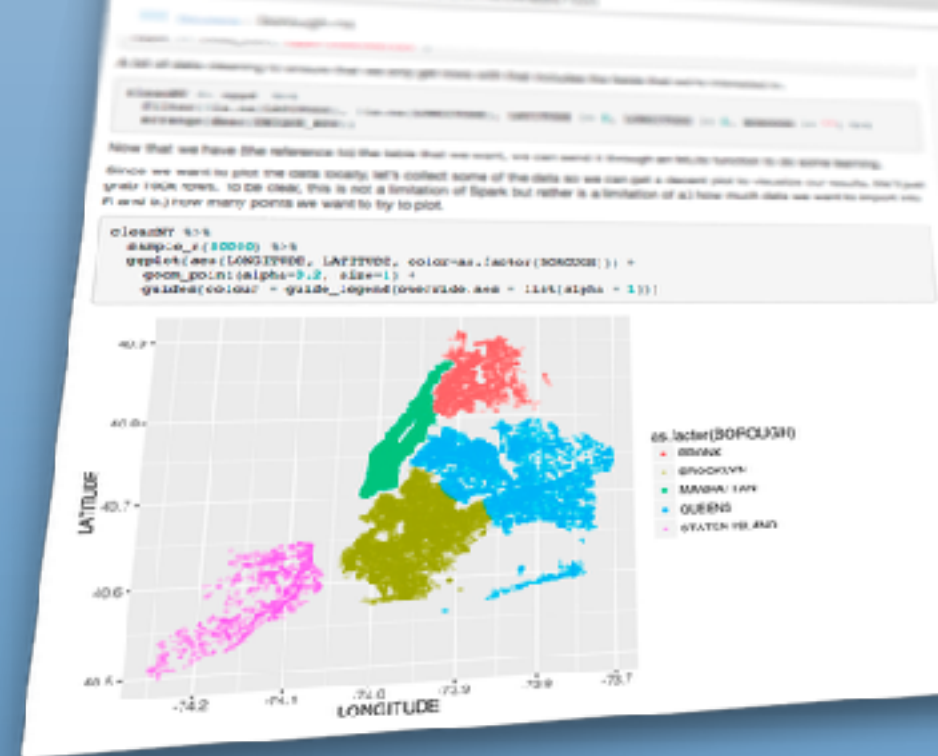
Manage Accounts...

- Push-button publishing from the RStudio IDE
- Manages all the content types you produce in R:
 - Shiny, R Markdown, plots, etc.
- On-premises
- Share data science artifacts

Connect Introductory Webinar

- Given March 17th
- Recording - <http://rstd.io/webinars>
- Slides - <http://rstd.io/rsc-webinar-1>

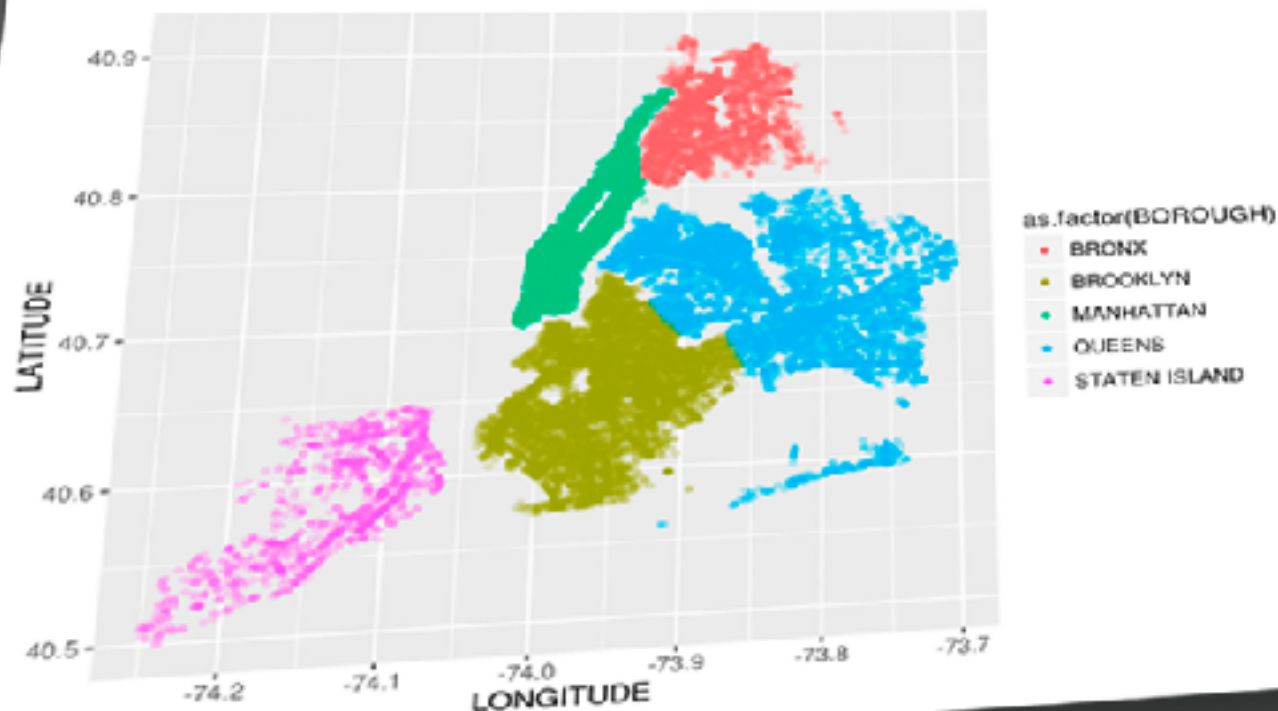
1. User management
2. Resource management
3. System security



Now that we have (the reference to) the table that we want, we can send it through an `mlb` function to do some learning.

Since we want to plot the data locally, let's collect some of the data so we can get a decent plot to visualize our results. We'll just grab 100k rows. To be clear, this is not a limitation of Spark but rather is a limitation of a.) how much data we want to import into R and b.) how many points we want to try to plot.

```
cleanNY %>%  
  sample_n(30000) %>%  
  ggplot(aes(LONGITUDE, LATITUDE, color=as.factor(BOROUGH))) +  
    geom_point(alpha=0.2, size=1) +  
    guides(colour = guide_legend(override.aes = list(alpha = 1)))
```



Who can view this document

Everyone

Who can change this document



Power User poweruser



John Doe johndoe

Find collaborator

Who runs this document on the server

The default user rstudio-connect

Administrators can create a custom "sanity" url to access this document. Your custom URL will be appended to your domain to form the complete path to your document.

/boroughs/

Your custom url:

<https://connect.rstudio.com/newyork/>

User Management

User Roles

- **Admin** - All privileges, but “irregular” actions are audited
- **Publisher** - Can upload new content.
- **Viewer** - Can see content.

Application Privileges

- **Collaborator** - Can publish new versions*, manage settings, refresh.
- **Viewer** - Can only see the existing content.

Connect Administrators

- Special admin-only actions
 - Admin tab (metrics, user management)
 - Application settings (vanity URLs, runas)
- Don't get everything for free
 - Must explicitly grant themselves privilege
 - These actions are audited

Default user role

- `[Authorization].DefaultUserRole`
- The role fresh users should have on the server
- Default is `publisher`

usermanager CLI

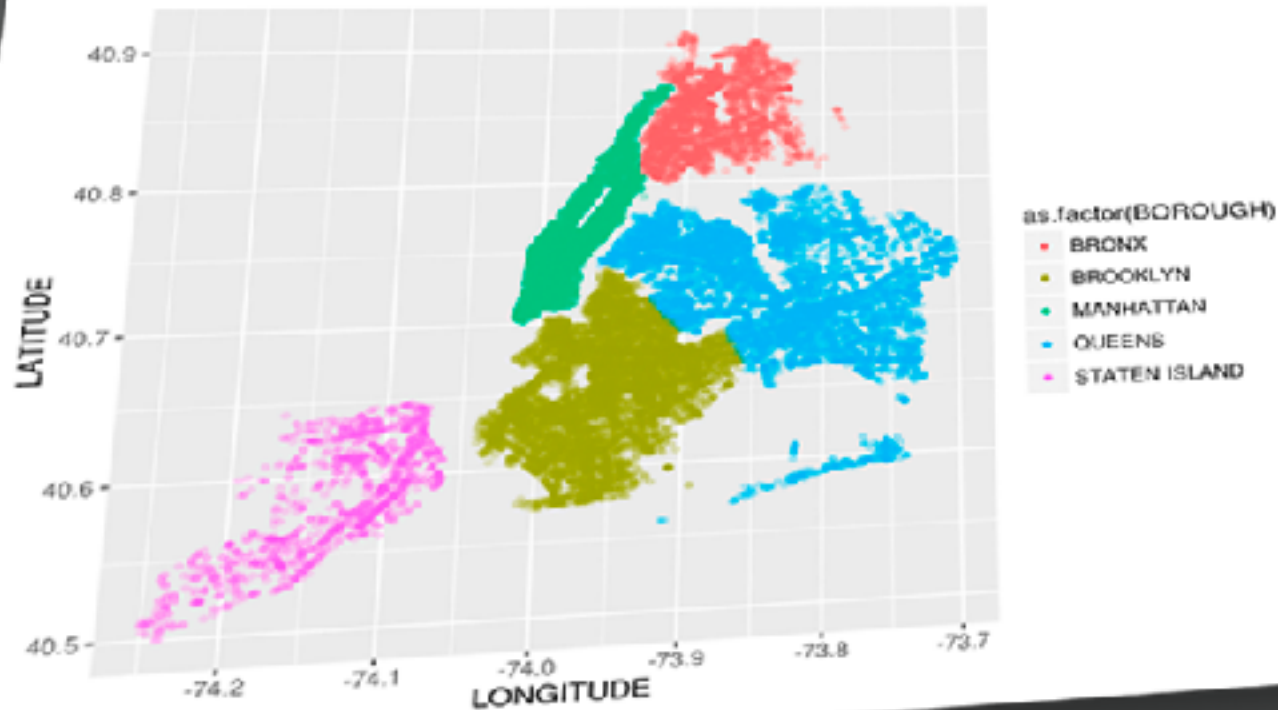
- Root-only command line interface to Connect
- Server must be stopped
- `/opt/rstudio-connect/bin/usermanager`
- `usermanager list`
- `usermanager alter`
- `usermanager audit -csv`

User Locking

- As of early 2017, no user deletion
- Can “lock” users
 - Forbids login/publishing
 - Does not count as a seat for licensing purposes
- Can rename users

Now that we have (the reference to) the table that we want, we can send it through an `mlb` function to do some learning. Since we want to plot the data locally, let's collect some of the data so we can get a decent plot to visualize our results. We'll just grab 100k rows. To be clear, this is not a limitation of Spark but rather is a limitation of a.) how much data we want to import into R and b.) how many points we want to try to plot.

```
cleanNY %>%  
  sample_n(30000) %>%  
  ggplot(aes(LONGITUDE, LATITUDE, color=as.factor(BOROUGH))) +  
    geom_point(alpha=0.2, size=1) +  
    guides(colour = guide_legend(override.aes = list(alpha = 1)))
```



Who can view this document

Everyone

Who can change this document

Power User poweruser

John Doe johndoe

Find collaborator

Who runs this document on the server

The default user rstudio-connect

Administrators can create a custom "sanity" url to access this document. Your custom URL will be appended to your domain to form the complete path to your document.

/boroughs/

Your custom url:

<https://connect.rstudio.com/newyork/>

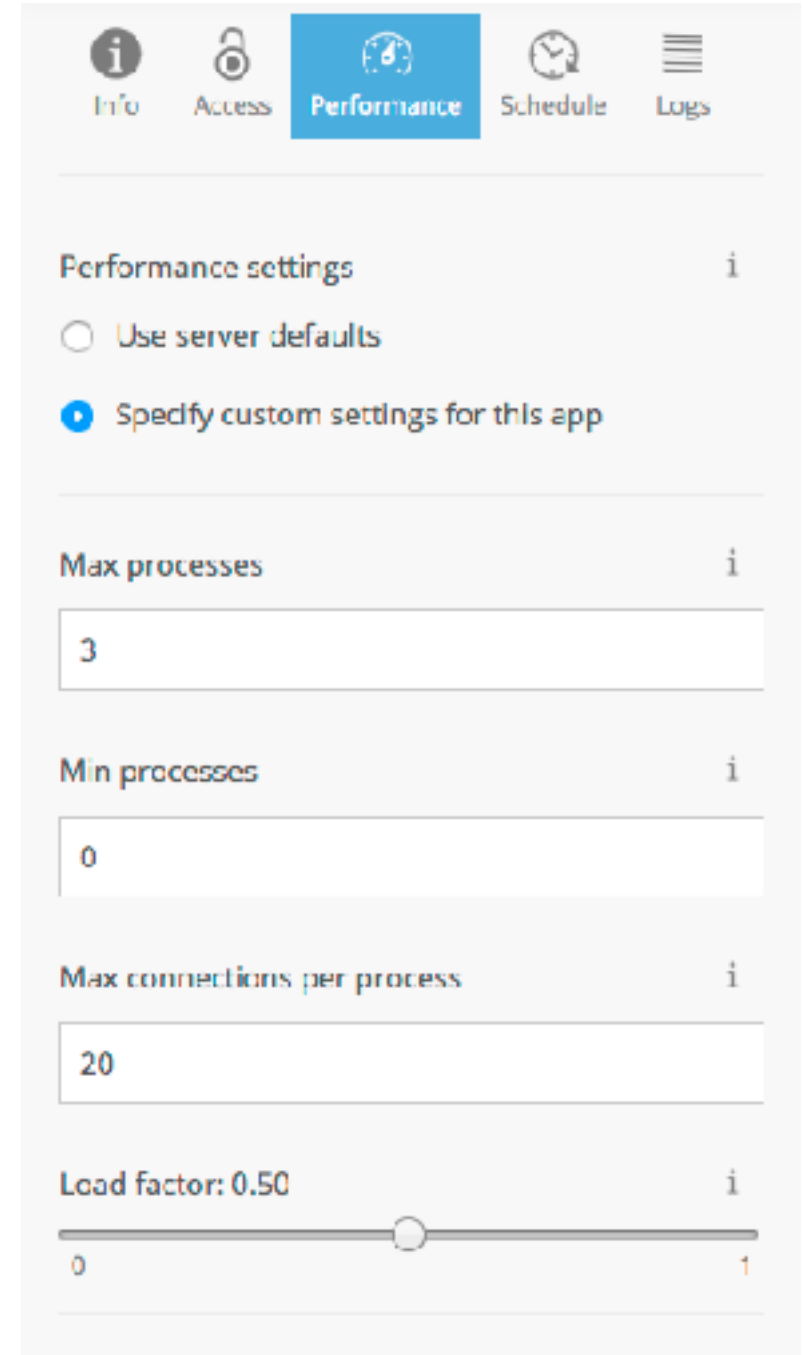
Resource Management

Resource Budgeting

- Resource requirements depend almost entirely on what your users deploy
- On-demand requests are serviced best-effort
- There are some knobs that let you cap resource requirements

Shiny Scaling

- Can scale Shiny applications to multiple processes
- Load-balance between these processes
- Min and max processes
- `[Scheduler].MinProcessLimit`



The image shows the 'Performance' settings tab in the Shiny application management interface. The top navigation bar includes 'Info', 'Access', 'Performance' (selected), 'Schedule', and 'Logs'. The 'Performance settings' section has two radio buttons: 'Use server defaults' (unselected) and 'Specify custom settings for this app' (selected). Below this, there are three input fields: 'Max processes' with a value of 3, 'Min processes' with a value of 0, and 'Max connections per process' with a value of 20. At the bottom, there is a 'Load factor' slider set to 0.50, with a range from 0 to 1. Each setting has an information icon (i) to its right.

Info Access **Performance** Schedule Logs

Performance settings *i*

☐ Use server defaults

☒ Specify custom settings for this app

Max processes *i*

3

Min processes *i*

0

Max connections per process *i*

20

Load factor: 0.50 *i*

0 1

Shiny Timeouts

- The maximum amount of time to wait for an app to start
 - `[Scheduler].InitTimeout = 60s`
- The minimum time to keep a worker process alive after it goes idle
 - `[Scheduler].IdleTimeout = 5s`

Report Concurrency

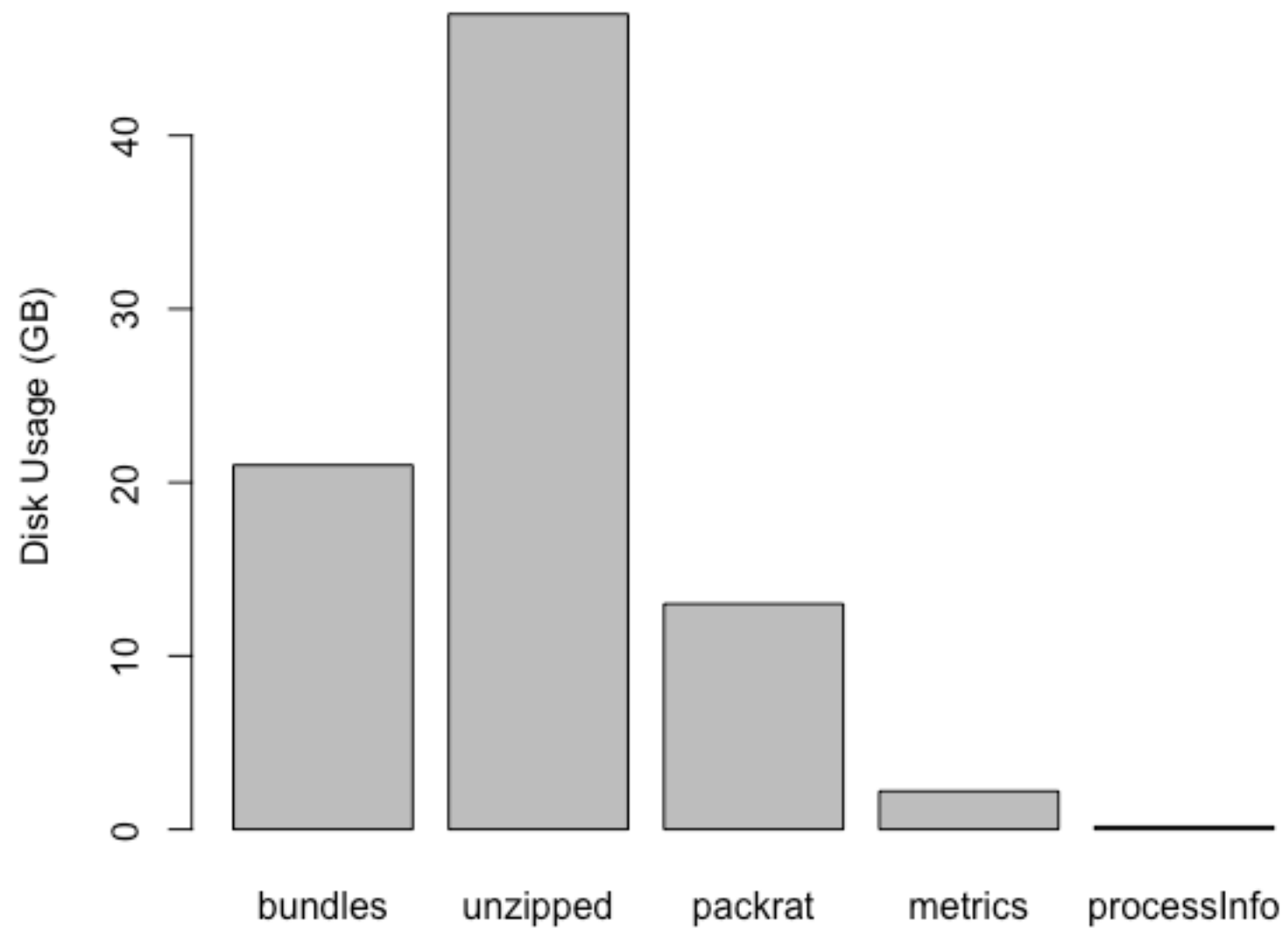
- `[Applications].ScheduleConcurrency`
- Maximum number of scheduled reports to run in parallel

Disk Usage

- Uploaded (compressed) bundles from users
- Unzipped bundles for running applications
- Packrat cache
 - One copy of each version of each package, specific to R version
- Metrics
- R process information/logs

Disk Usage

- >2k content
- ~85GB



Bundle Retention

- `[Applications].BundleRetentionLimit`
- Default is 0, meaning preserve all bundles forever
- If experiencing problems with large bundles, instruct users to minimize the size of their bundles
 - Provision data on the server separately

Process Info (“Jobs”) Retention

- `[Jobs].MaxCompleted = 100`
 - **≤ 100 process logs per application**
- `[Jobs].OldestCompleted = 720h`
 - **Process logs older than 30 days**
- **Once either constraint is violated, the job will be deleted from disk**

IT Considerations

- `[Server.DataDir] = /var/lib/rstudio-connect`
- If you can keep bundle sizes small, disk usage should be small (<100GB)
- Fast disk (SSD) will improve performance
- NFS is fine
 - Consider performance
 - SQLite database **must** be on local disk

Backups

- Include:
 - `/var/lib/rstudio-connect`
 - `/etc/rstudio-connect`
 - Database, if stored outside of the data directory
- Stop the server, backup, start the server

R Versions

- Support multiple versions of R
- Scanning
 - Check common locations and PATH
- Or register explicit R locations
- Multiple version matching algorithms available
- <http://rstd.io/rsc-r>

Process Management - Supervisors

- Prefix to the R process invocation
- `[Applications].Supervisor = nice -n 2`
- Can be used to provision resources before spawning R
- <http://rstd.io/rsc-supervisors>

Process Management - RunAs

- `[Applications].RunAs`
- `rstudio-connect` **user**, by default
- **Primary group:** `rstudio-connect`
- <http://rstd.io/rsc-runas>

Customizing RunAs per Application

- Configured in the dashboard
- Only RStudio Connect admins can modify
- Must be a user in the primary group of the default RunAs user (`rstudio-connect`)
- Enables you to make special system resources available to particular applications

Current User Execution

- `[Applications].RunAsCurrentUser`
- Defaults to false (disabled)
- Requires PAM authentication
- Allows you to execute a Shiny application as the Unix account of the current viewer
- <http://rstd.io/rsc-runascurrentuser>

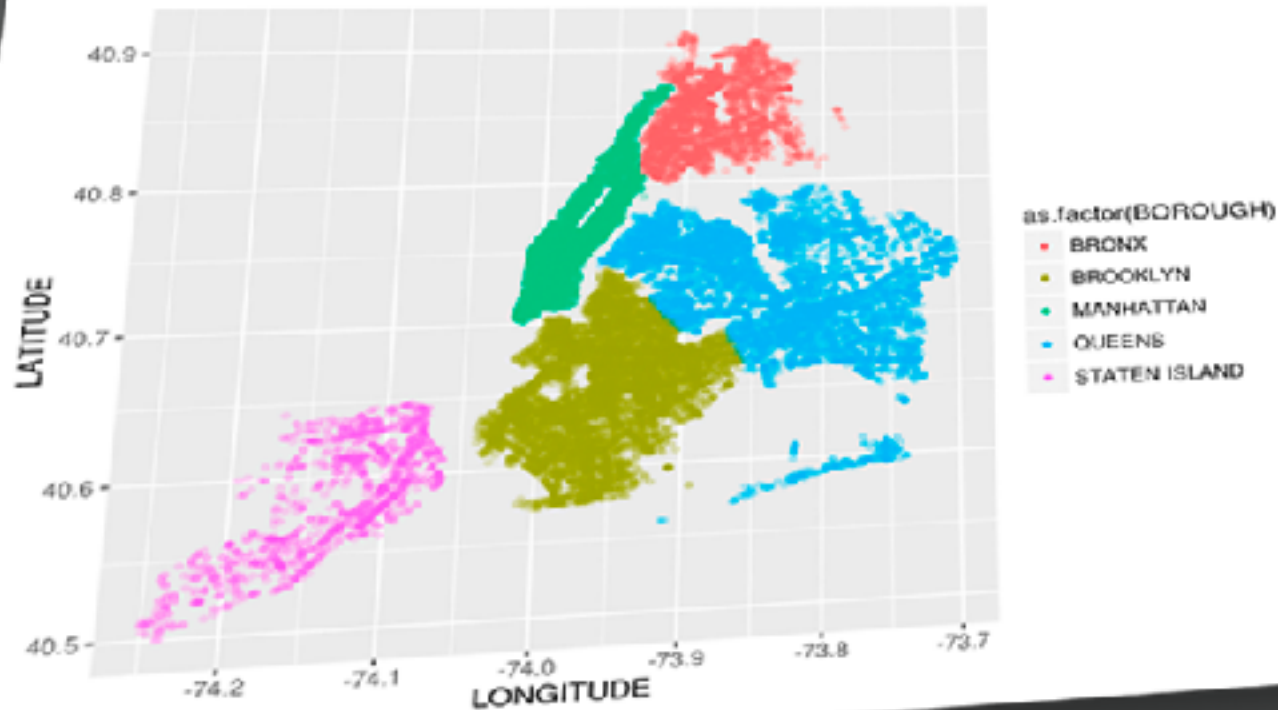
PAM Sessions

- Use PAM to establish the environment and resources available to R sessions
- Specify a custom PAM service

Now that we have (the reference to) the table that we want, we can send it through an `mlr3` function to do some learning.

Since we want to plot the data locally, let's collect some of the data so we can get a decent plot to visualize our results. We'll just grab 100k rows. To be clear, this is not a limitation of Spark but rather is a limitation of a.) how much data we want to import into R and b.) how many points we want to try to plot.

```
cleanNY %>%  
  sample_n(30000) %>%  
  ggplot(aes(LONGITUDE, LATITUDE, color=as.factor(BOROUGH))) +  
    geom_point(alpha=0.2, size=1) +  
    guides(colour = guide_legend(override.aes = list(alpha = 1)))
```



System Security

HTTPS

- If accepting passwords, you should be using TLS
- `[Https].Listen/Key/Certificate`
- `[HttpRedirect].Listen/Target`
- Forward HTTP traffic to HTTPS

Browser Security

- Options available to control...
 - HSTS
 - ContentTypeSniffing
 - X-Frame-Options
 - Custom headers
- <http://rstd.io/rsc-browser-security>
- <https://www.ssllabs.com/ssltest/>

Audit Logs

- Monitor for system changes
- Audits user/group management, application settings, bundle downloads, etc.
- Dump to CSV using `usermanager`

Download Bundle

Bundle



**Connect
Staging**



**Connect
Production**



Staging Environments

Messages for Users

- Custom HTML messages for users
- `[Server].PublicWarning`
 - Visible on unauthenticated landing page
- `[Server].LoggedInWarning`
 - Visible above recent content when logged in

Private Packages

- Packrat gracefully handles CRAN, Bioconductor, public source repositories
- Private packages require a private CRAN
- Host your packages on your own CRAN mirror behind a firewall
- <http://rstd.io/rsc-private-repos>

Additional resources

- Download & 45-day free trial: <http://rstd.io/rsc>
- Admin Guide: <http://rstd.io/rsc-admin>
- IT Q&A: <http://rstd.io/rsc-it-qa>
- Authentication details: <http://rstd.io/rsc-auth>
- Release notes: <http://rstd.io/rsc-news>