

Programmieren mit Statistischer Software - Simulationen am Server -

Moritz Berger, Cornelia Oberhauser

Institut für Statistik, LMU München

SS 2015

Gliederung

- 1 Verbindung mit dem Netzlaufwerk
- 2 Verbindung zum Server
- 3 Navigation auf dem Server
- 4 Batch-Jobs

Gliederung

- 1 Verbindung mit dem Netzlaufwerk
- 2 Verbindung zum Server
- 3 Navigation auf dem Server
- 4 Batch-Jobs

Verbindung mit dem Netzlaufwerk (im CIP-Pool)

Es ist möglich, sich von Außen mit dem Netzlaufwerk im CIP-Pool zu verbinden.

- Jederzeit Zugriff auf alle dort gespeicherten Daten
- Dateien können direkt von dort geöffnet, bearbeitet und wieder dort gespeichert werden. (Nur etwas langsamer.)
- Siehe auch CIP-Wiki:
`http://gforge.stat.uni-muenchen.de/gf/project/cip/wiki/?pagename=FileServerZugriff`

Schritte zur Verbindung mit dem Netzlaufwerk

- 0) Von außerhalb des LRZ-Netzwerks: Verbindung mit dem LRZ-Netzwerk aufbauen (VPN-Client)
- 1) Windows Explorer öffnen → Computer → Netzlaufwerk verbinden (oben in der Menüleiste)
- 2) 1. Dialogbox: Welcher Netzwerkordner soll zugeordnet werden?
 - Laufwerk: (noch nicht verwendeten) Laufwerksbuchstabe wählen
 - Ordner: `\\hotsun2.stat.uni-muenchen.de\benutzername`
 - Option *Verbindung mit anderen Anmeldeinformationen herstellen* auswählen
- 3) 2. Dialogbox: Netzwerkkenwort eingeben
 - Benutzername: **CIPWAP\benutzername** (identifiziert Domäne und Nutzer)
 - Kennwort: eigenes Kennwort

Trennen des Netzlaufwerks

- Im Windows Explorer:
Rechtsklick auf das Laufwerk → Trennen

Gliederung

- 1 Verbindung mit dem Netzlaufwerk
- 2 Verbindung zum Server**
- 3 Navigation auf dem Server
- 4 Batch-Jobs

SSH-Programm

- Alle Statistik-Hauptfachstudenten haben mit ihrer CIP-Kennung Zugriff auf die bereitgestellten Server
- Siehe auch CIP-Wiki:
`http://gforge.stat.uni-muenchen.de/gf/project/cip/wiki/?pagename=Simulationen+am+Server`
- Verbindung mit den Servern im CIP-Pool über den SSH Secure Shell Client / SSH-Client Putty

Für den Zugriff von zu Hause:

- Eine VPN-Verbindung zum LMU-Netz aufbauen.

Siehe auch: [http:](http://www.lrz.de/services/netz/mobil/vpn/anyconnect/)

[//www.lrz.de/services/netz/mobil/vpn/anyconnect/](http://www.lrz.de/services/netz/mobil/vpn/anyconnect/)

- SSH-Client Putty (einmalig) installieren

Seite des LRZ:

<https://www.lrz.de/services/security/putty/>

Download:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

→ Datei putty.exe herunterladen und ausführen

Zugriff auf den Server (SSH Secure Shell / Putty)

- Zur Verfügung stehende Server: cipserv1, cipserv3
- Host Name: *servername.stat.uni-muenchen.de*
- User Name und Password: eigene CIP-Rechner Kennung
- Beim ersten Zugriff kann das eigene Profil unter beliebigem Namen gespeichert werden
- Nach Verbinden mit dem Server befindet man sich direkt in seinem Ordner auf dem Netzlaufwerk des Servers *hotsun2.stat.uni-muenchen.de* und kann beliebig darin navigieren und sich ansehen welche Dateien dort liegen.

Gliederung

- 1 Verbindung mit dem Netzlaufwerk
- 2 Verbindung zum Server
- 3 Navigation auf dem Server**
- 4 Batch-Jobs

Nützliche Befehle (allgemein)

- `$ > ls`: Auflistung der Dateien (inklusive Ordner) im aktuellen Arbeitsverzeichnis
- `$ > cd`: Wechsel des Arbeitsverzeichnisses
 - `$ > cd ordnername`: Wechsel in den Unterordner ordnername
 - `$ > cd ..`: Wechsel in den übergeordneten Ordner
- `$ > mkdir ordnername`: make directory (Anlegen eines neuen Ordners ordnername)
- `$ > rmdir ordnername`: remove directory (Löschen des Ordners ordnername)

- Hinweis: Das Tastenkürzel um Text einzufügen ist auf dem Server [Umschalt] + [Einfg].

Nützliche Befehle (zur Verwendung von R)

- `$ > R`: Starten von R

Auf dem Server kann ganz normal in R gearbeitet werden. Notwendige Pakete können auch wie gewohnt mit `install.packages()` installiert werden. Mit `q()` kann R beendet werden.

Gliederung

- 1 Verbindung mit dem Netzlaufwerk
- 2 Verbindung zum Server
- 3 Navigation auf dem Server
- 4 Batch-Jobs**

Starten von abgeschlossenen Jobs

R-Programme können mit

```
$ > nohup R CMD BATCH dateiname.R &
```

gestartet werden.

- nohup: Job wird nicht abgebrochen, wenn die Verbindung zum Server getrennt wird
- &: Nach Eingabe wird der Fokus wieder direkt an die Konsole zurückgegeben

Verwaltung von Prozessen auf dem Server

- `$ > top`: Auflistung der laufenden Prozesse auf dem Server
Jeder Prozess hat eine eindeutige Nummer (PID). Angezeigt wird u.a. der User (USER), die abgelaufene Zeit und das Programm (R) des Prozesses. Mit `q` kommt man wieder zurück in die Konsole!
- `$ > kill prozessnummer`: Abbrechen von laufenden Prozessen

Aufbau des Programms

- Das R-Programm muss selbstständig und fehlerfrei lauffähig sein.
- Das Arbeitsverzeichnis kann nicht mit `setwd()` festgelegt werden. Als Arbeitsverzeichnis dient immer der Ordner in dem man sich auf dem Server gerade befindet.
- Alle nötigen Funktionen mit `source()` und alle nötigen Pakete mit `library()` laden.
- Alle Graphiken mit `pdf() ... dev.off()` und alle Objekte mit `save()` speichern. (mit relativen Pfaden arbeiten!)

Beachte: Alles was nicht explizit gespeichert wird, ist nach Beendigung des Jobs nicht mehr da!

Output

- Der Output des Jobs wird im aktuellen Arbeitsverzeichnis in einer Datei names *dateiname.Rout* gespeichert.
- Entspricht beim Ausführen des R-Programms dem Output in der Console.
- Zusätzlich wird am Ende, falls das Programm fehlerfrei gelaufen ist, die abgelaufene Zeit angegeben.
- Die Datei kann mit jedem beliebigen Editor betrachtet werden.
- Auf dem Server mit `$ > cat dateiname.Rout` einsehbar.