

# Anleitung für den R-Server

Softwareversion RStudio-Server Pro 1.0.44

von Sven Urban  
Stand: 18.1.2017

## Inhaltsverzeichnis

Allgemeine Einführung .....	2
Starten der Verbindung .....	2
Dateien zum Server hochladen .....	4
Dateien vom Server herunterladen.....	5
Auswahl der R-Version .....	6
Packages installieren .....	9
Installation eines Benutzer-Paketes .....	9
Zentrale Installation eines Paketes .....	12
Session Management .....	13
Session starten .....	13
Session wechseln/anzeigen.....	14
Session beenden .....	15
Abmelden/Beenden.....	16
Zusammenarbeit an einem Projekt.....	17
Ein Projekt freigeben .....	17
Projekt als Teilnehmer öffnen .....	18
Aktive Teilnehmer erkennen .....	19
Ressourcenverwaltung .....	20
R-Server und lokales R .....	21

## Allgemeine Einführung

Der *R-Studio-Server Professional*, nachfolgend wird die Bezeichnung *R-Server* sowohl für die Hard- als auch die Software als Gesamtsystem benutzt, bietet eine Entwicklungsumgebung zur Statistiksoftware „R“.

Ergänzende Tips und Anregungen sind eingerückt und in kursiver Schrift gehalten.

*Diese Anleitung geht davon aus, daß der Leser mit der Bedienung von R-Studio grundlegend vertraut sei; es werden nur diejenigen Aspekte behandelt, die über die Desktop-Version hinausgehen.*

Es handelt sich hierbei um eine *Client-Server-Architektur*, d.h. der lokale Arbeitsplatz wird nur zur Ein- und Ausgabe verwendet, alle Berechnungen finden jedoch auf dem Server statt. Ebenso liegen das Programm „R“ samt seiner installierten Pakete sowie die Scripte der Anwender zentral auf dem Serversystem.

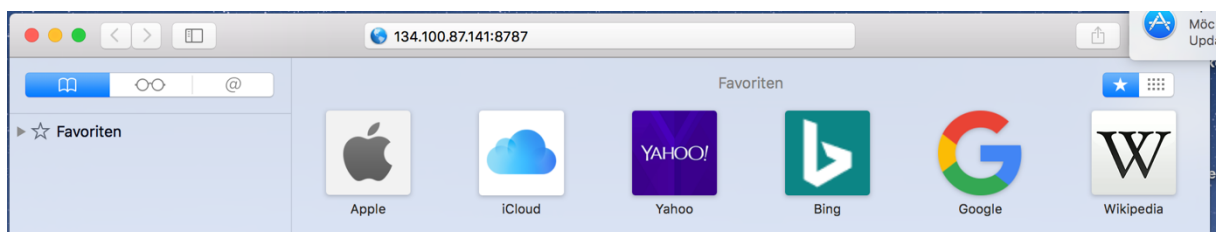
Da alle Rechenleistung vom Server erbracht wird, kann das Clientsystem auch leistungsschwach sein, z.B. ein Tablettcomputer.

Getestet wurden folgende Betriebssystem/Browser-Kombinationen:

- Windows/Mozilla
- Mac OSX/Safari

*Viele weitere Kombinationen sind wahrscheinlich möglich, wurden aber bisher noch nicht ausgetestet.*

## Starten der Verbindung

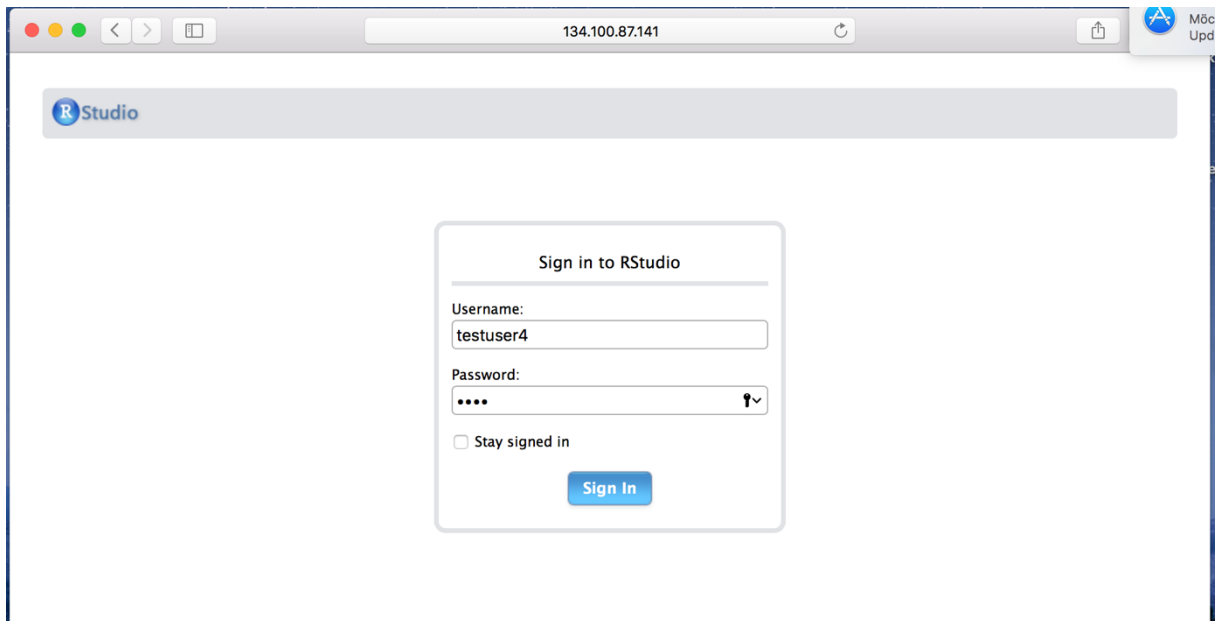


In einem Standardbrowser wird die IP-Adresse des RStudio-Servers eingegeben: die Adresse lautet **134.100.87.141:8787**

Wichtig hierbei ist, daß die Portnummer **8787** nach einem Doppelpunkt eingegeben werden muß! Der R-Server ist nämlich kein Standard-Webserver und benötigt auch keinen als Basis. Im Prinzip könnte auf den R-Server weltweit zugegriffen werden, allerdings kann das eingeschränkt werden.

*Hinweis: Unter Windows wird von der Verwendung des Internet Explorers dringend abgeraten, da dieser erfahrungsgemäß bei Java und JavaScript gerne Kompatibilitätsprobleme hat. Stattdessen sollte der Firefox von Mozilla verwendet werden.*

Es erscheint der Anmeldebildschirm:

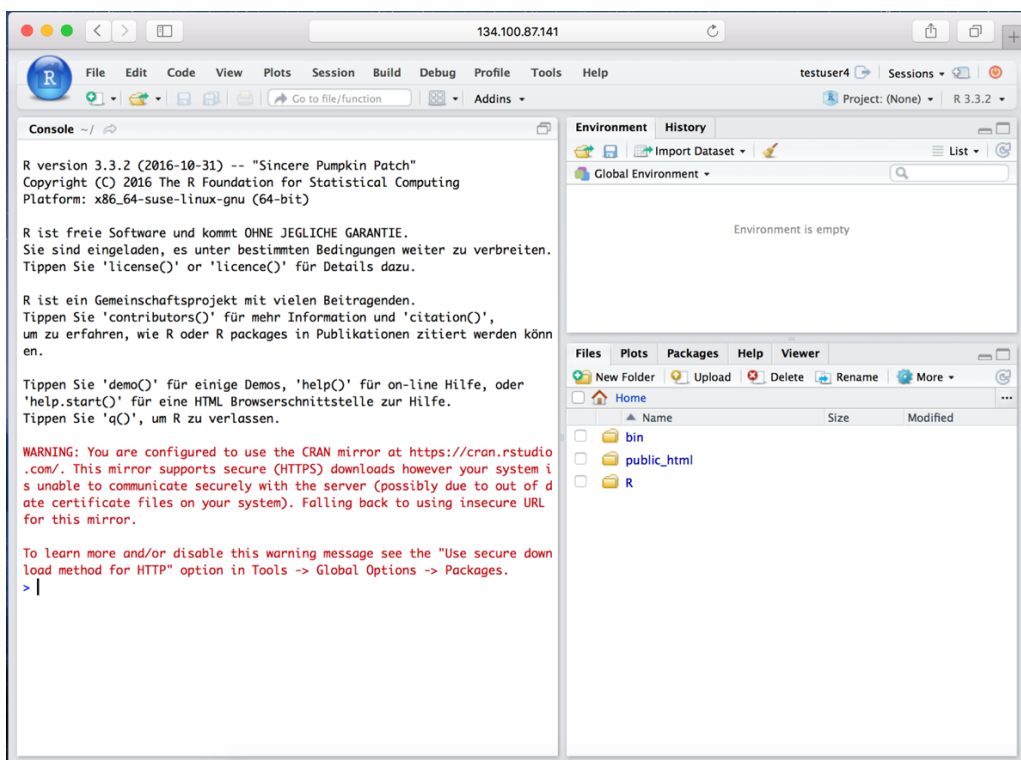


Hier sind nun der Benutzername und das Paßwort einzugeben.

*Jeder Benutzer benötigt eine Kennung. Masterstudenten und Mitarbeiter erhalten eine persönliche Kennung; Kursteilnehmer fragen bitte den jeweiligen Dozenten.*

*Das Paßwort ist nur für dieses System und derzeit mit keinem anderen Paßwort an der UniHH synchronisiert. Das Paßwort kann durch den Benutzer derzeit nicht geändert werden.*

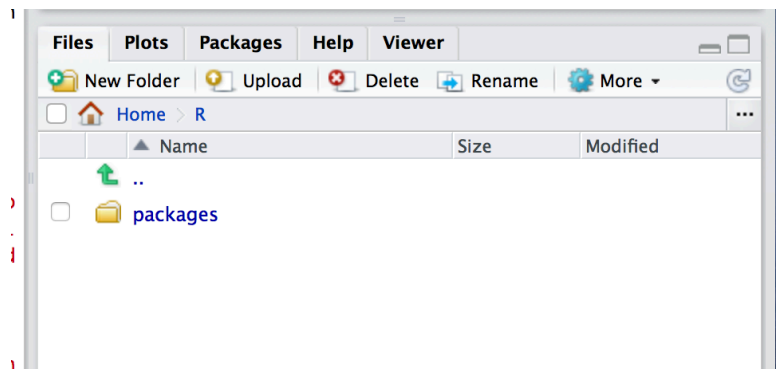
Es erscheint der Hauptbildschirm:



Dieser Bildschirm sieht dem „normalen“ R-Studio zum Verwechseln ähnlich. Und er funktioniert – von einigen wenigen Ausnahmen abgesehen – auch genau so. Jeder, der schon einmal mit R-Studio gearbeitet hat, wird sich sofort zurechtfinden.

Eine Besonderheit ist, daß sich die Dateien unter *Files* nicht auf dem Rechner befinden, vor dem man sitzt, sondern auf dem R-Server. D.h. man hat seine Dateien immer dabei.

*Files* zeigt das persönliche Homeverzeichnis, auf welches nur der Benutzer selbst Zugriff hat. Das Dateisystem ist am Anfang nahezu leer, aber es können Dateien oder auch Ordner in einer beliebigen Strukturtiefe erzeugt werden.

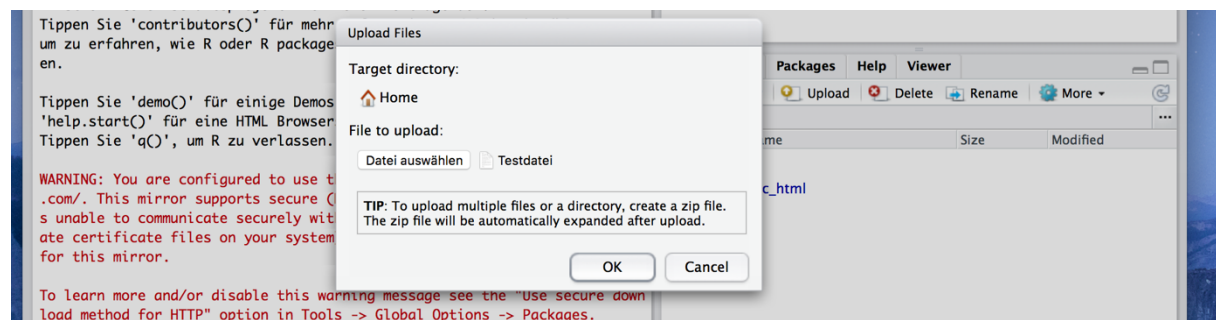


Im Ordner *R* befindet sich ein Unterordner *packages*. Hier wird der R-Server die Packages installieren, die der Benutzer selbst heruntergeladen und installiert hat.

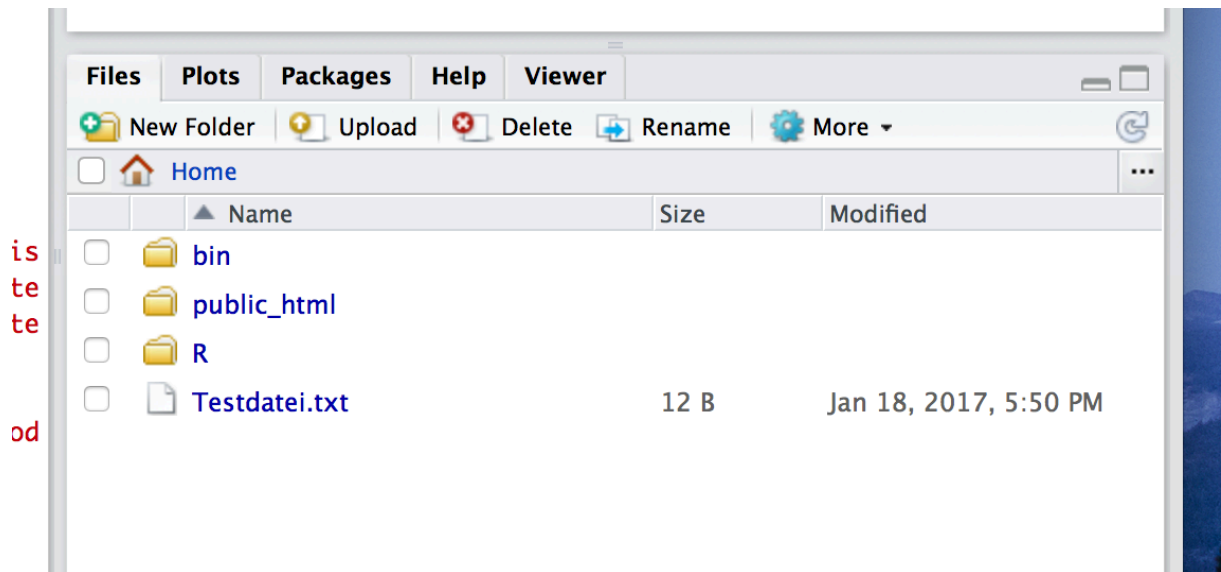
*Es ist nicht empfehlenswert, von Hand in dem Ordner „packages“ Dateien zu kopieren, zu löschen oder zu bearbeiten. Diese Aufgabe sollte dem R-Server überlassen bleiben.*

## Dateien zum Server hochladen

Selbstverständlich lassen sich Dateien vom Arbeitsrechner in das persönliche Homeverzeichnis des Servers ganz einfach hochladen. Hierfür wird im Fenster *Files* einfach auf den Button *Upload* geklickt.



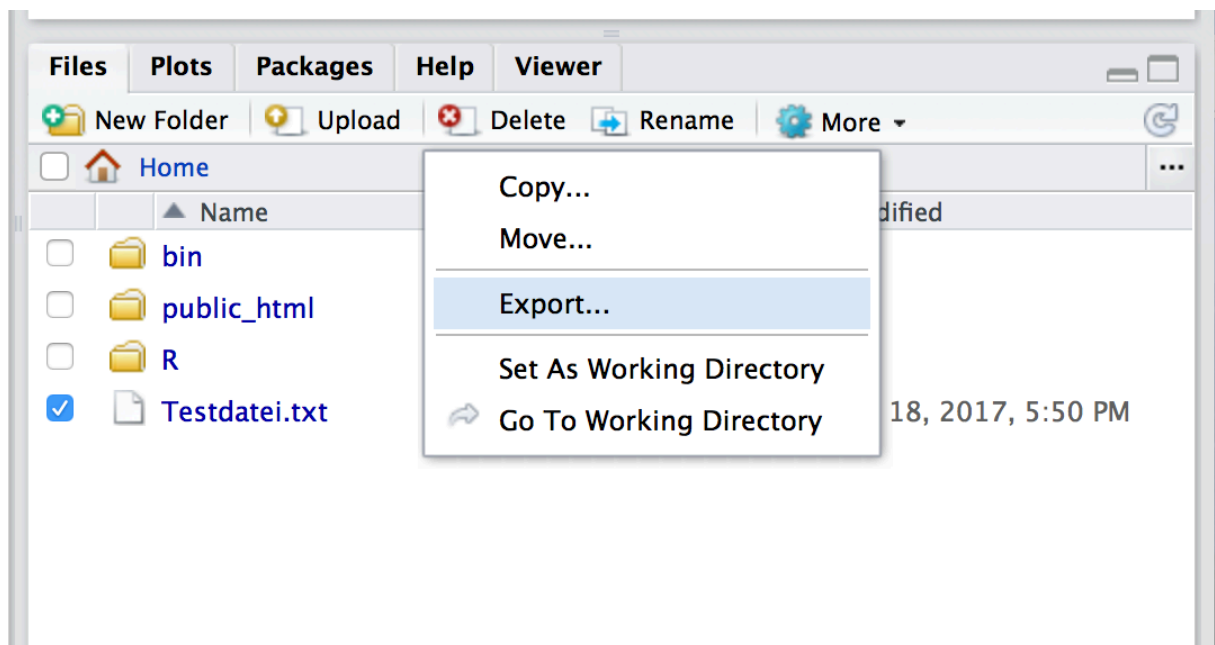
Nun wird mit dem Button *Datei auswählen* die entsprechende Datei ausgewählt. Als Ziel ist automatisch das eigene Homeverzeichnis voreingestellt. Mit *OK* wird die Datei kopiert.



Die Datei „Testdatei.txt“ wurde erfolgreich hochgeladen und erscheint nun im Fenster.

## Dateien vom Server herunterladen

Ebenso einfach können Dateien vom Server auf den lokalen Arbeitsplatzrechner heruntergeladen werden.



Zunächst wird die entsprechende Datei markiert, indem das Kästchen angeklickt wird. Der Download wird unter R-Server als „Export“ bezeichnet und befindet sich im Menü unter *More*.

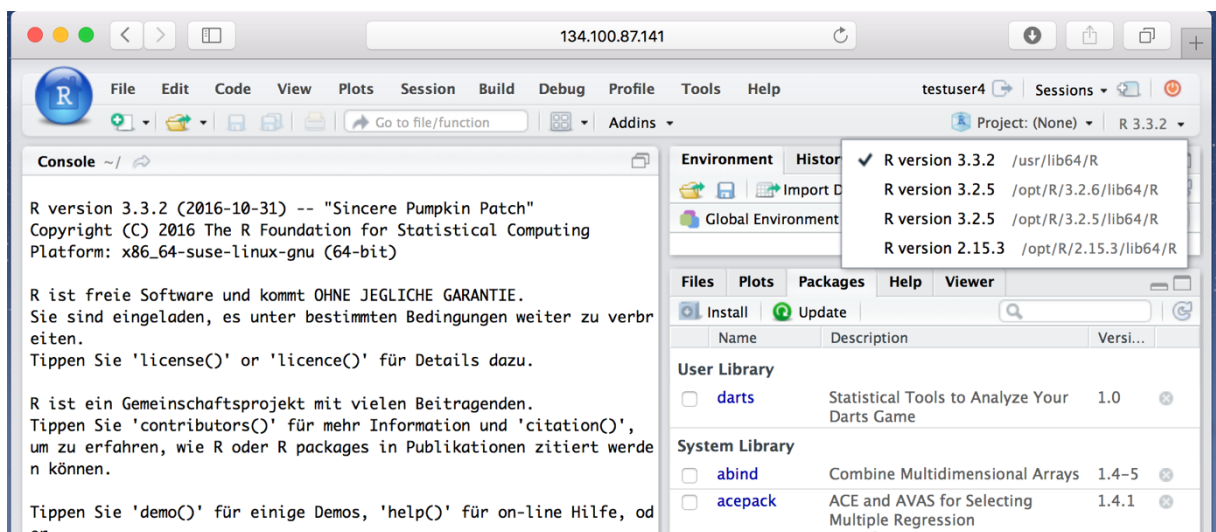


Die Datei wurde erfolgreich heruntergeladen. Hier hat das Betriebssystem die Datei auch gleich geöffnet.

## Auswahl der R-Version

Eine Besonderheit des RStudio-Server Pro ist die Möglichkeit, verschiedene R-Versionen parallel benutzen zu können. Diese müssen aber aus den Quellen selbst kompiliert werden, was entsprechend zeitraubend ist.

In der rechten oberen Ecke des Hauptfensters wird die aktuell verwendete R-Version angezeigt.

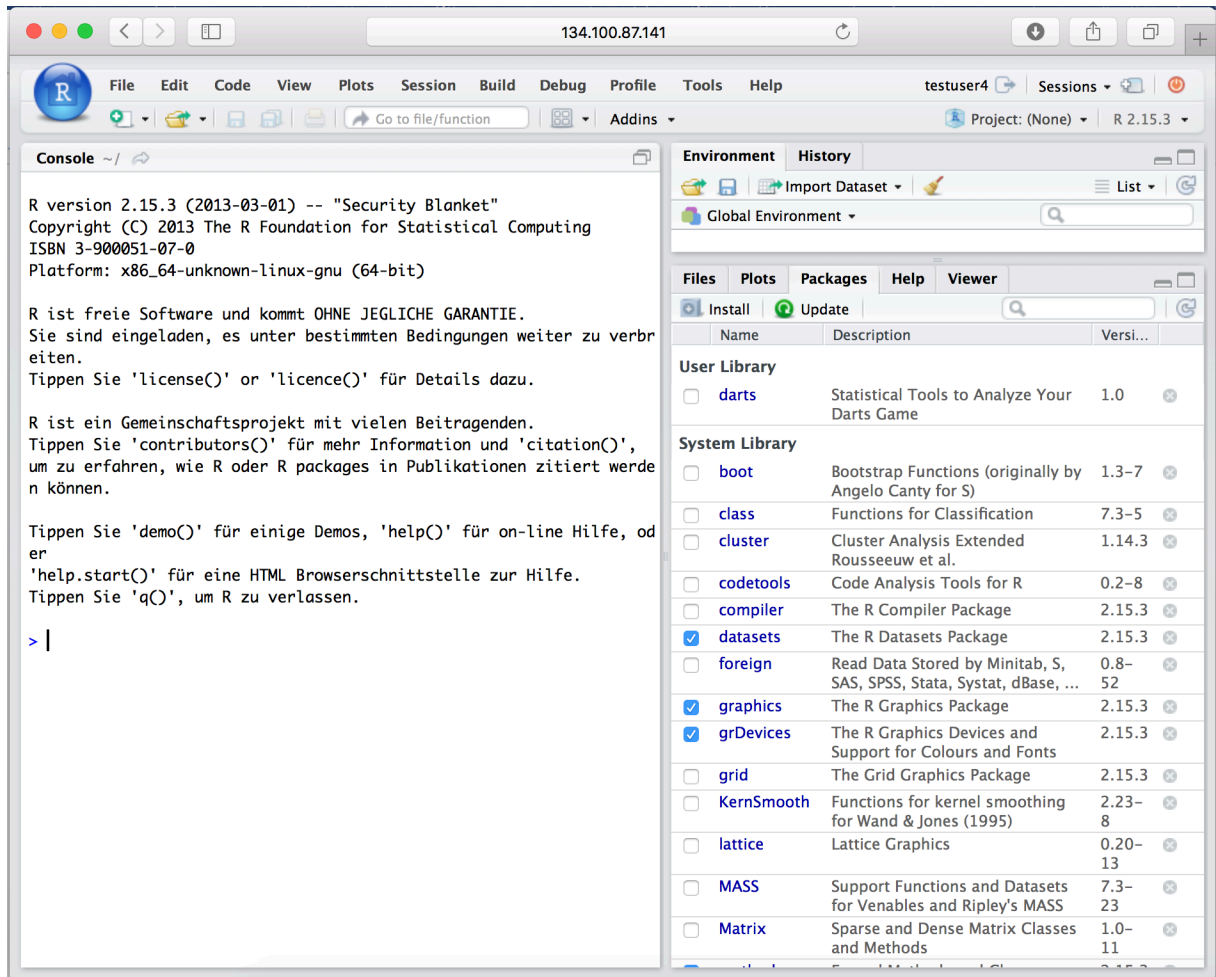


Ein Klick auf den Pfeil zeigt die verfügbaren Versionen an, sowie die Lage ihrer Hauptverzeichnisse.

*Derzeit ist die 3.3.2 die Hauptversion. Bei einem Update kann die Hauptversion auf einen neueren Stand gebracht werden, ohne daß die*

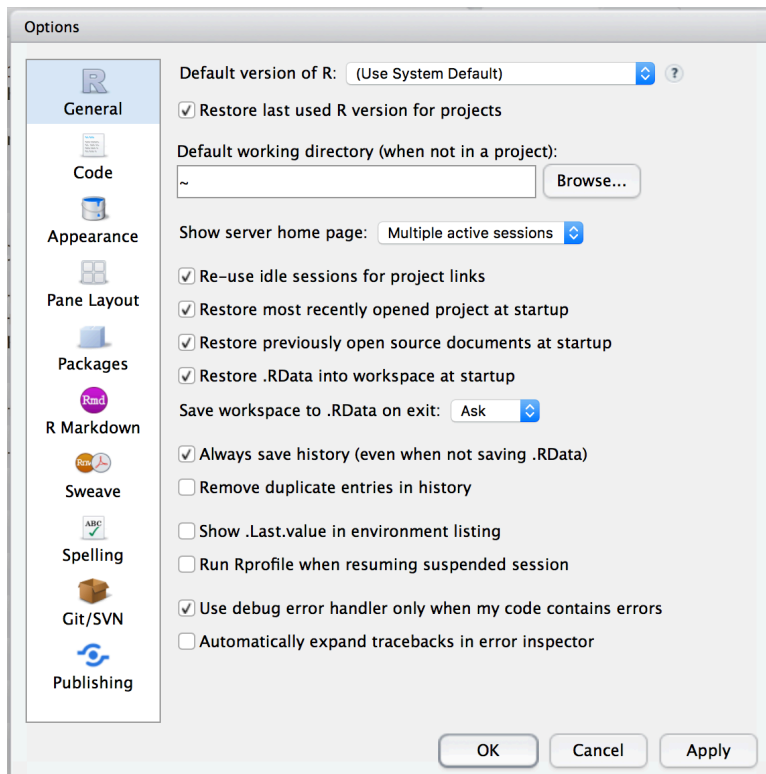
Pakete neu installiert werden müssen. Allerdings kann man sich nicht darauf verlassen, daß die Hauptversion auf genau diesem Versionsstand bleibt.

Durch einfaches Klicken mit der Maus kann die Version innerhalb von Sekunden gewechselt werden.



Hier wurde auf die Version 2.15.3 umgestellt.

Selbstverständlich kann die Auswahl auch permanent getroffen werden. Unter *Tools* den Punkt *Global Options* wählen. Die Auswahlmöglichkeiten befinden sich im Bereich unter „General“.



Die Voreinstellung (*Use System Default*) bedeutet, daß der Anwender stets die Version verwendet, die beim R-Server als Hauptversion eingestellt ist. Natürlich lassen sich die Voreinstellungen während der Benutzung jederzeit auch manuell überschreiben.

*Manche Änderungen der Voreinstellungen werden erst bei der nächsten Anmeldung wirksam.*

## Packages installieren

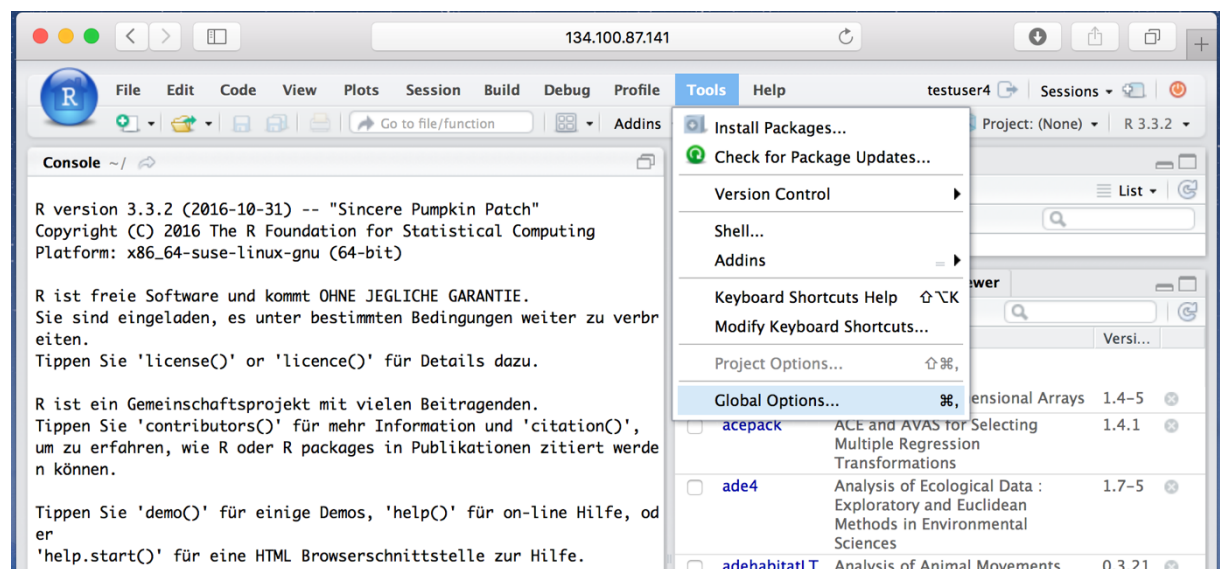
Inzwischen werden mehr als 9000 Pakete für R angeboten, die leider nicht alle auf dem R-Server installiert sein können, weil hierfür der Plattenplatz nicht ausreichen würde.

Bei der Installation von Paketen müssen grundsätzlich zwei Typen unterschieden werden:

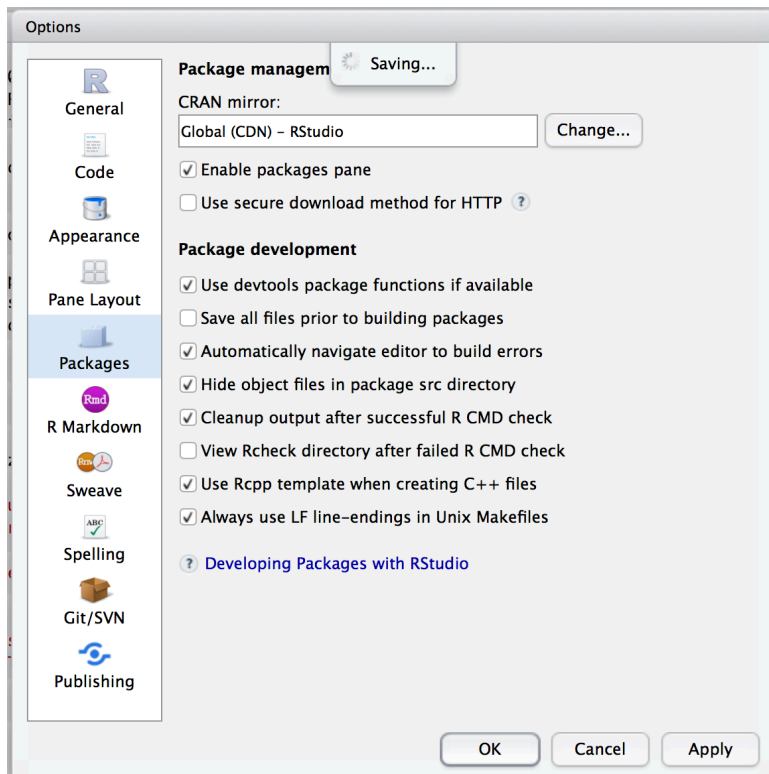
- 1 Pakete nur für den Benutzer, der sie installiert hat, und
- 2 Pakete für alle Benutzer, die zentral installiert werden.

*Im Falle von Paketen, die für Unterrichtszwecke benötigt werden, wird dringend empfohlen, daß der Dozent diese zunächst unter seiner persönlichen Kennung installiert und testet (Typ 1), bevor die Anforderung an den Administrator weitergeleitet wird. Die anschließende Installation vom Typ 2 bietet jedoch den enormen Vorteil, daß die gewünschten Pakete sofort für alle Benutzer – und damit auch die Teilnehmer des Kurses - verfügbar sind.*

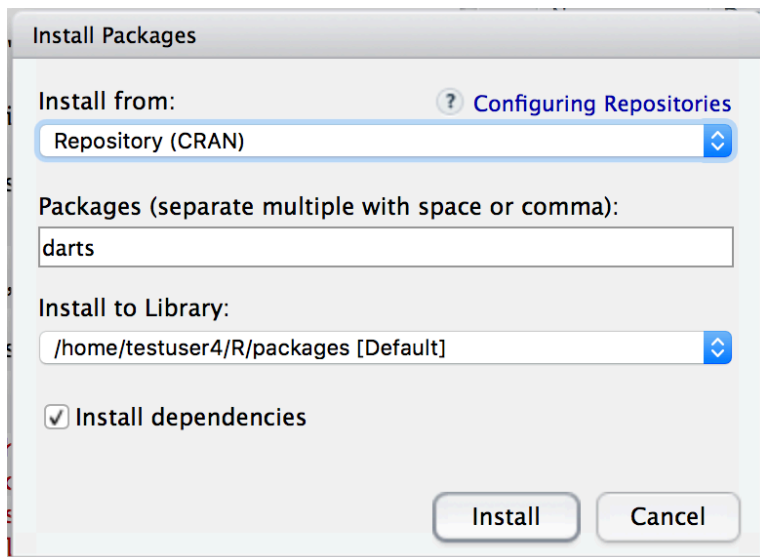
## Installation eines Benutzer-Paketes



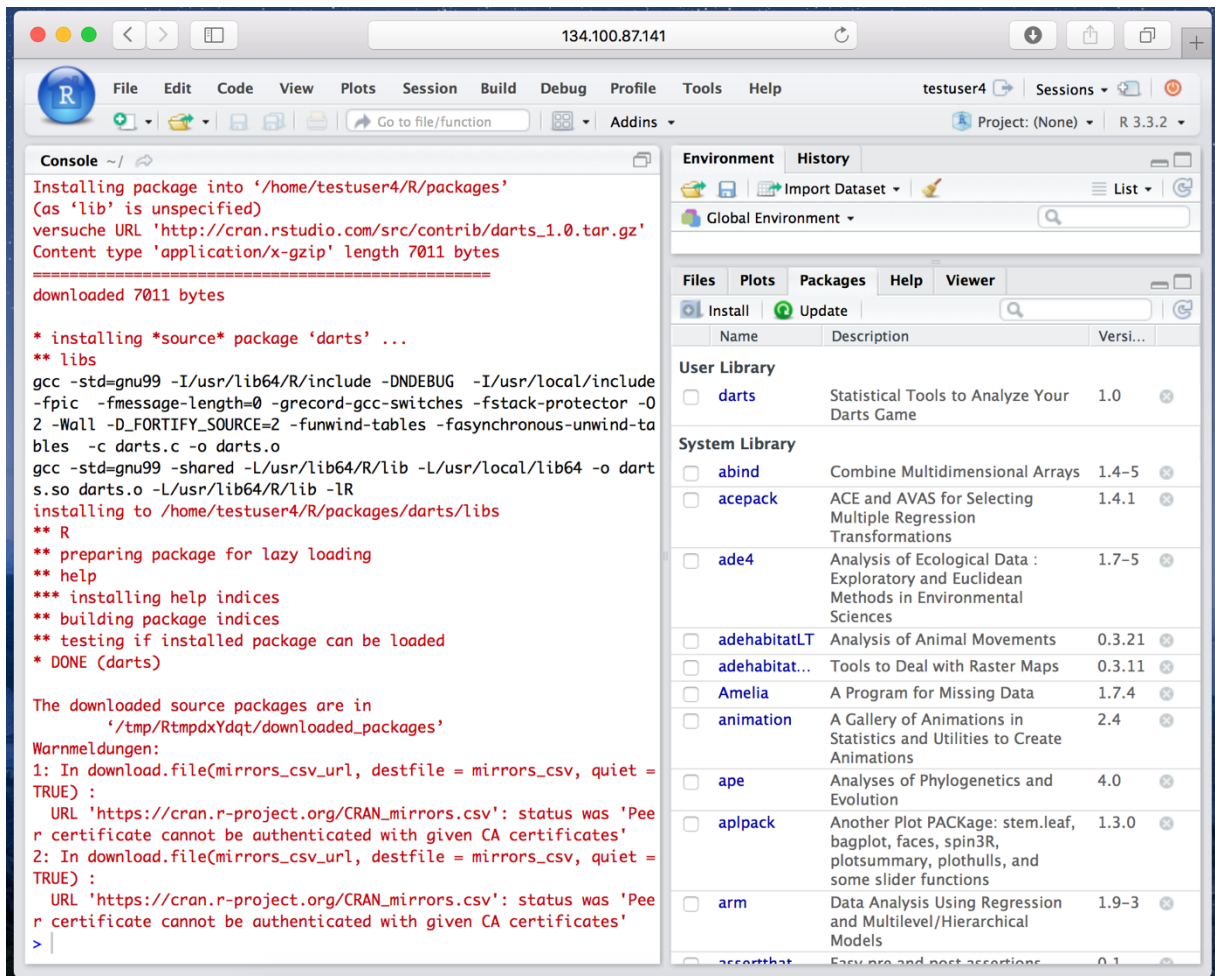
Zunächst sind einige Vorbereitungen zu treffen. Unter *Tools* wird wieder *Global Options* gewählt, um einige grundlegende Voreinstellungen im Bereich „Packages“ zu treffen.



**Ganz wichtig** ist, daß der Haken für *Use secure download method for http* entfernt wird! So dann ist ein CRAN-Mirror zu wählen. Hier bietet sich die Uni Münster an, weil die innerhalb des DFN liegt und damit der Download entsprechend schnell ist.



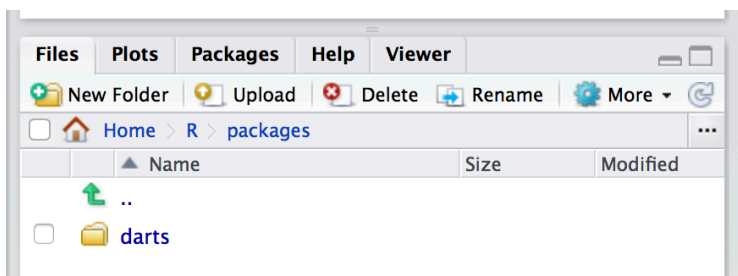
Hier soll das Paket „darts“ installiert werden. Liegt das Paket schon als Datei vor, so kann dies unter *Install from* entsprechend ausgewählt werden.



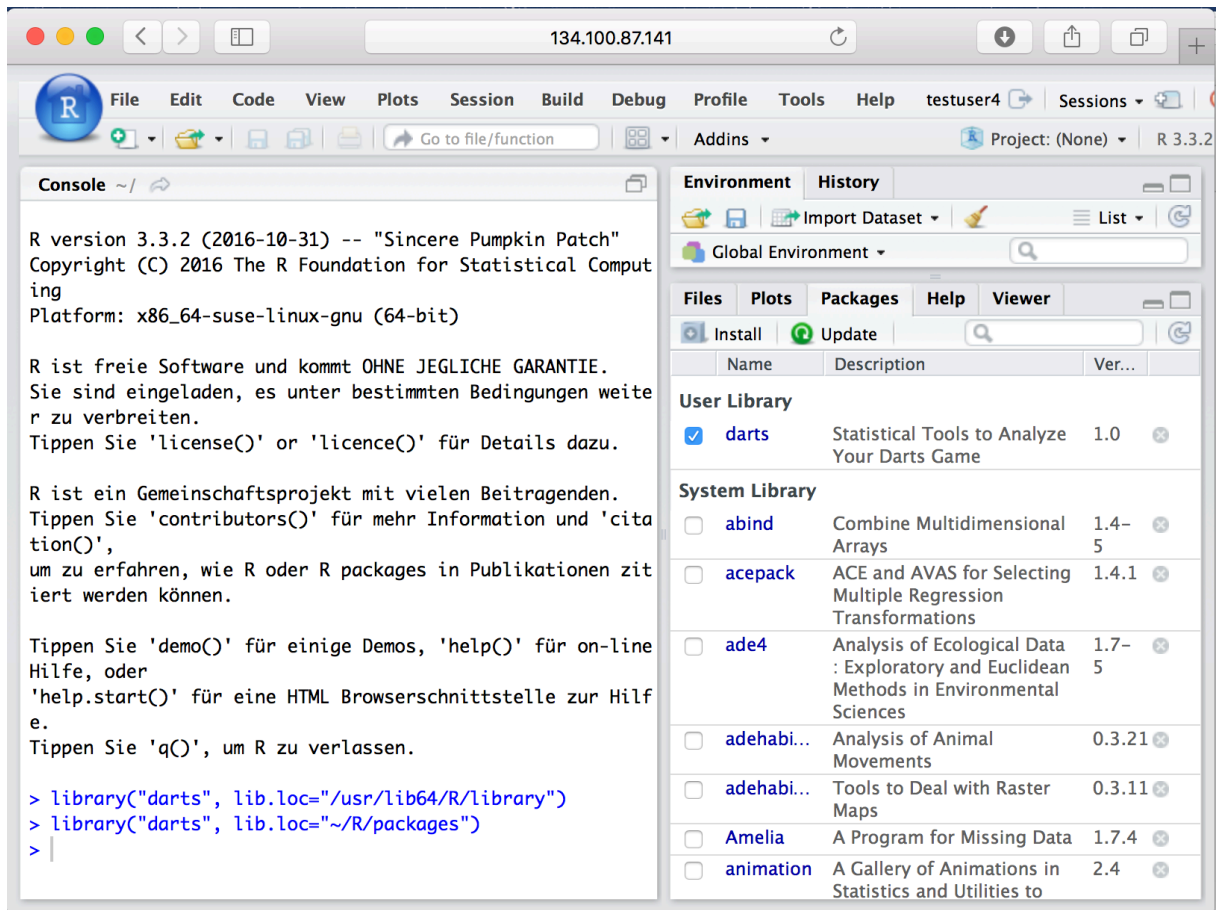
Im Fenster *Console* kann man sehen, wie das Paket heruntergeladen und anschließend kompiliert wird. Die Zeile `* DONE (darts)` zeigt an, daß es erfreulicherweise geklappt hat. Im Fenster *Packages* ist das neue Paket unter *User Library* zu sehen.

*Vom Benutzer als User Library installierte Pakete können in allen R-Versionen verwendet werden.*

*Das bedeutet aber nicht, daß die Pakete in allen Versionen von R auch funktionieren, denn das ist von den Paketen selbst abhängig!*



Im Fenster *Files* können wir im Ordner *R/packages* sehen, daß das neue Paket *darts* hier installiert wurde.



Die Pakete lassen sich wie in R üblich verwenden. Auf dem hier gezeigten System ist das Paket „darts“ in zwei Versionen vorhanden. Die obere Zeile lädt das zentral gespeicherte Paket und die untere Zeile das durch den Benutzer installierte Paket.

*Es kann vorkommen, daß eine ganz bestimmte (z.B. ältere) Version eines Paketes benötigt wird. Auf diese Weise kann die Verwendung der richtigen Version sichergestellt werden.*

## Zentrale Installation eines Paketes

Pakete vom Typ 2 können derzeit nicht über die grafische Oberfläche installiert werden, sondern nur durch den Administrator. Pakete vom Typ 2 müssen zudem für jede gewünschte R-Version separat installiert werden; das ist zwar umständlich, ermöglicht aber unterschiedliche Paket-Versionen für unterschiedliche R-Versionen.

## Session Management

Für den R-Server ist *Session* ein zentraler Begriff. Eine Session ist sozusagen eine einzelne Programmausführung von R, der Server kann eine Vielzahl von Sessions parallel ausführen, da er über leistungsstarke Ressourcen verfügt. Jede Session verwendet ihren eigenen Speicherbereich und nutzt auch Kopien der Bibliotheken, so daß keine Interaktion zwischen den einzelnen Sessions stattfindet.

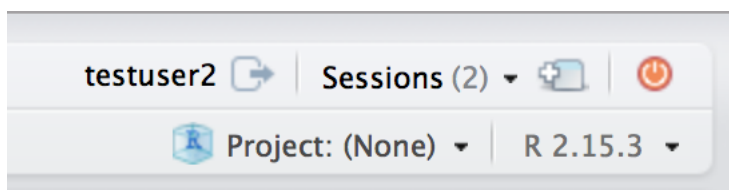
Startet ein Benutzer jedoch mehrere Sessions, so muß er selbst darauf achten, daß diese nicht Probleme beim Einlesen der Daten oder beim Abspeichern der Ergebnisse bekommen, wenn auf dieselben physischen Dateien zugegriffen wird.

Eine Session läuft auf dem R-Server weiter, auch wenn der Benutzer das Clientsystem abschaltet oder die Verbindung abbricht. Das ist sehr nützlich, weil der Benutzer mit seinem Clientsystem ohne Belastung durch R weiterarbeiten kann, während gleichzeitig der Server – quasi im Hintergrund – komplexe Berechnungen durchführt.

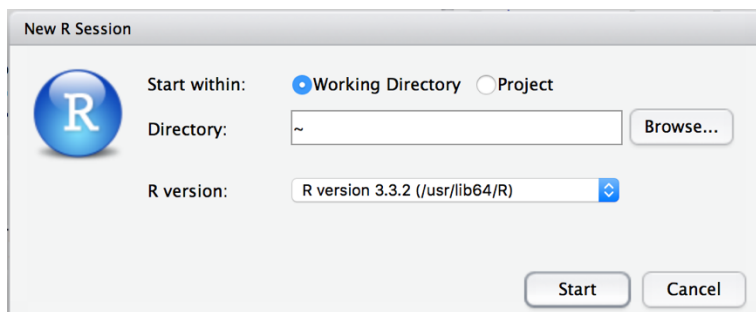
*Es ist wichtig, daß die Benutzer Sessions, die nicht mehr gebraucht werden, auch wirklich beenden, weil nur so alle Ressourcen wieder freigegeben werden.*

### Session starten

Die erste Session wird automatisch beim ersten Anmelden gestartet. Je nach Voreinstellungen „erkennt“ der Server auch eine bereits früher gestartete Session und stellt diese automatisch wieder her.



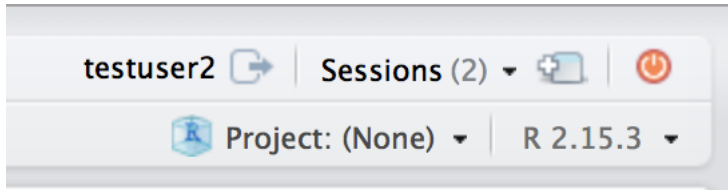
Eine weitere Session wird gestartet, indem auf das Symbol  geklickt wird.



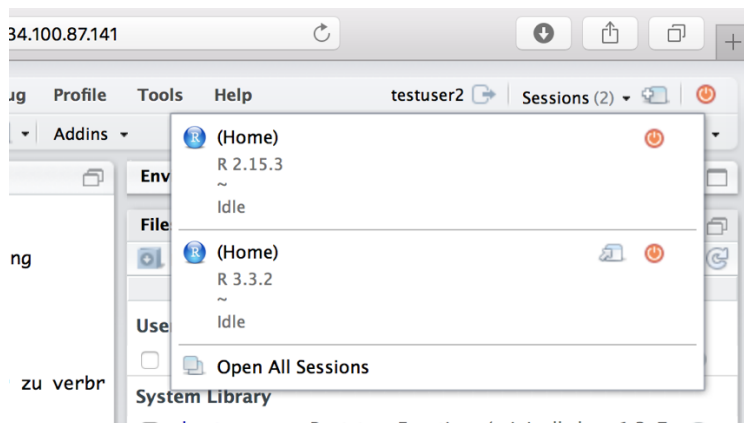
Jetzt kann die R-Version gewählt werden, allerdings ist eine Änderung auch später noch jederzeit möglich.

Dann kann noch gewählt werden, ob ein Verzeichnis oder ein Projekt geöffnet werden soll. Mit einem Klick auf *Start* wird dann die neue Session gestartet und geöffnet.

## Session wechseln/anzeigen



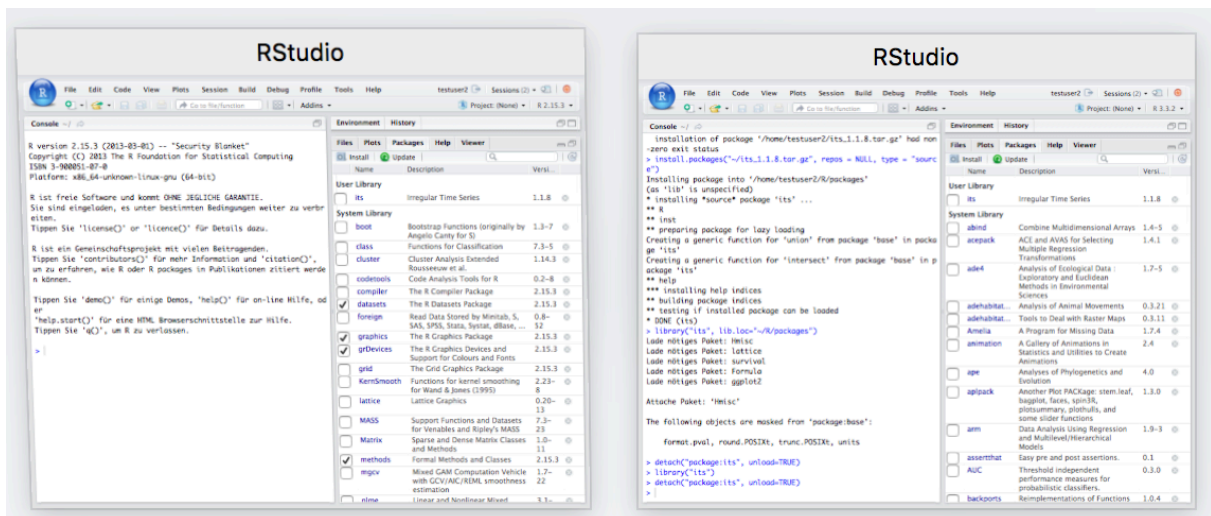
Um zu einer anderen Session zu wechseln, wird in der oberen rechten Ecke auf den Pfeil hinter Session geklickt. Dem Benutzer wird jederzeit die Anzahl seiner Sessions angezeigt.



Hier hat der Benutzer zwei Sessions gleichzeitig gestartet. Die oberste ist immer die, die gerade im Client angezeigt wird. Hier sieht man auch schön, daß der Benutzer in den Sessions unterschiedliche R-Versionen gleichzeitig nutzt. Eine andere Session wird nun durch einfaches Klicken in die gewünschte Session ausgewählt.




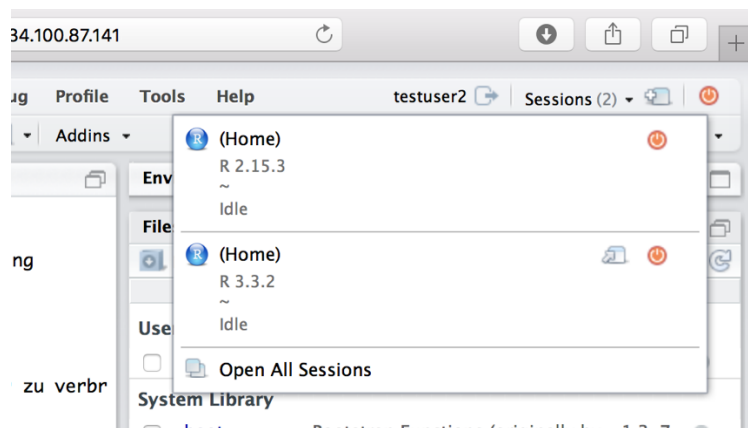
Wird jedoch das Symbol angeklickt, so öffnet sich die Session in einem weiteren Fenster.




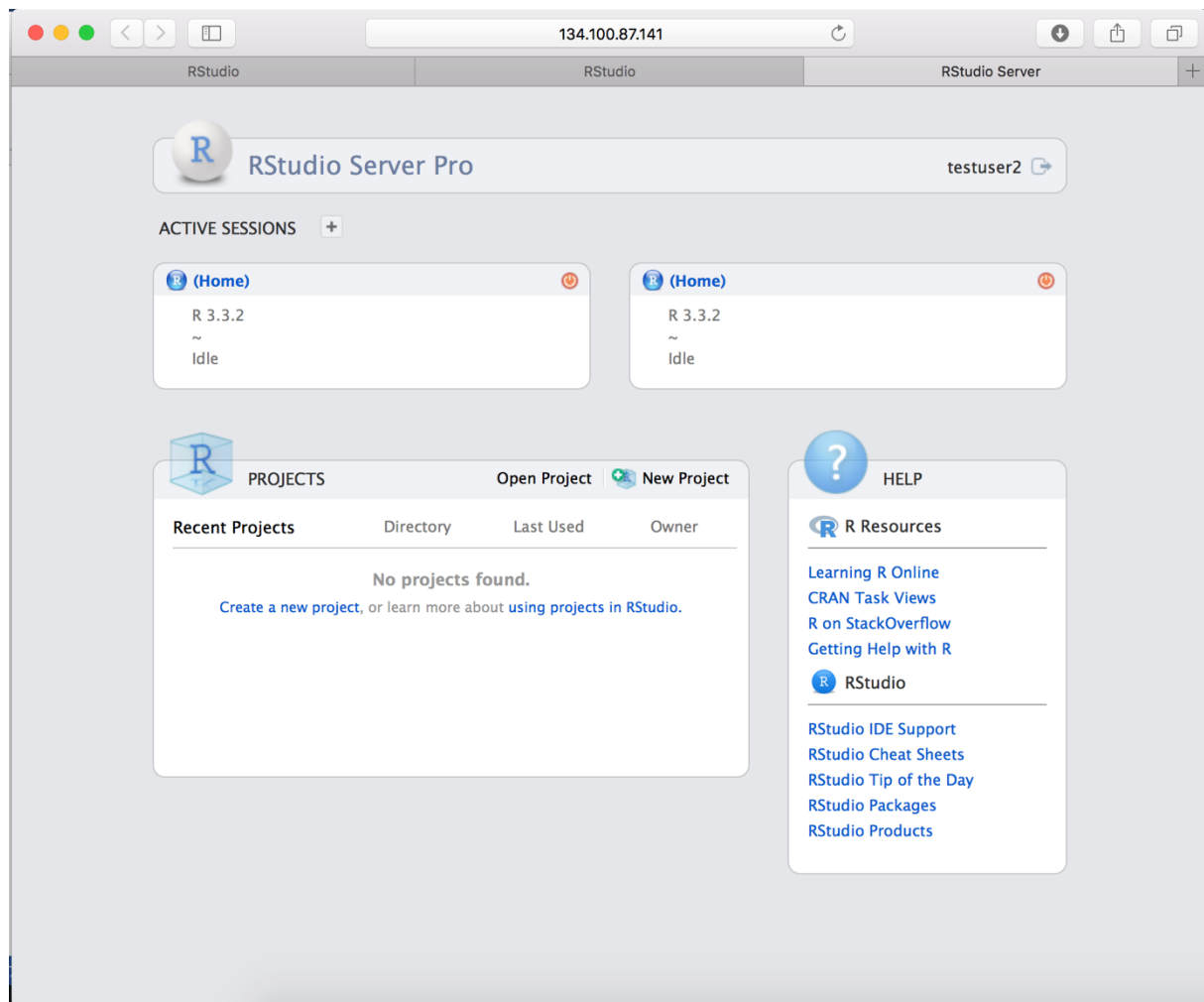
## Session beenden

Wie gesagt, sollten nicht mehr benötigte Sessions beendet werden, um alle Ressourcen des Servers wieder für andere Benutzer verfügbar zu machen.

Die derzeit angezeigte Session kann jederzeit mit einem Klick auf das Symbol  beendet werden. Hat der Benutzer noch weitere aktive Sessions, so landet er auf dem Startbildschirm.




In der Session-Liste kann jede Session mit einem Klick auf das Symbol  beendet werden. Hat der Benutzer noch weitere aktive Sessions, so landet er auf dem Startbildschirm.



Hier kann er z.B. zu einer seiner weiteren Sessions wechseln, oder auch eine neue Session starten.

*Es ist nicht möglich, sich eine Session in mehr als einem Fenster gleichzeitig anzeigen zu lassen. Der R-Server erkennt den Konflikt und suspendiert jeweils eine der Verbindungen.*

## Abmelden/Beenden

Der angemeldete Benutzer kann sich jederzeit vom R-Server abmelden, indem das Symbol  angeklickt wird. Alternativ kann auch gleich das Browserfenster geschlossen werden. Hierbei werden jedoch alle Sessions abgemeldet.

*Die Abmeldung läßt jedoch alle Sessions im Hintergrund weiterlaufen. Eine Abmeldung bedeutet nicht, eine Beendigung der Sessions.*

*Vergleichbar wäre dies mit dem Abschalten des Bildschirms bei einem PC, der trotzdem weiterläuft und Berechnungen durchführen kann.*

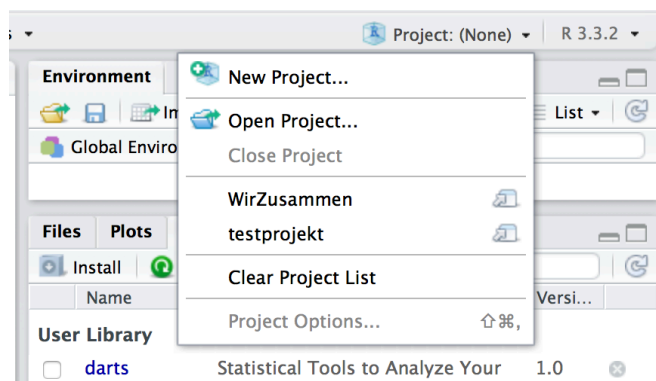
## Zusammenarbeit an einem Projekt

Eine der wichtigsten Funktionen von R-Server ist die Möglichkeit, daß mehrere Benutzer gemeinsam an einem Projekt arbeiten können. Diese Möglichkeit gibt es bei der „normalen“ Version von R-Studio nicht.

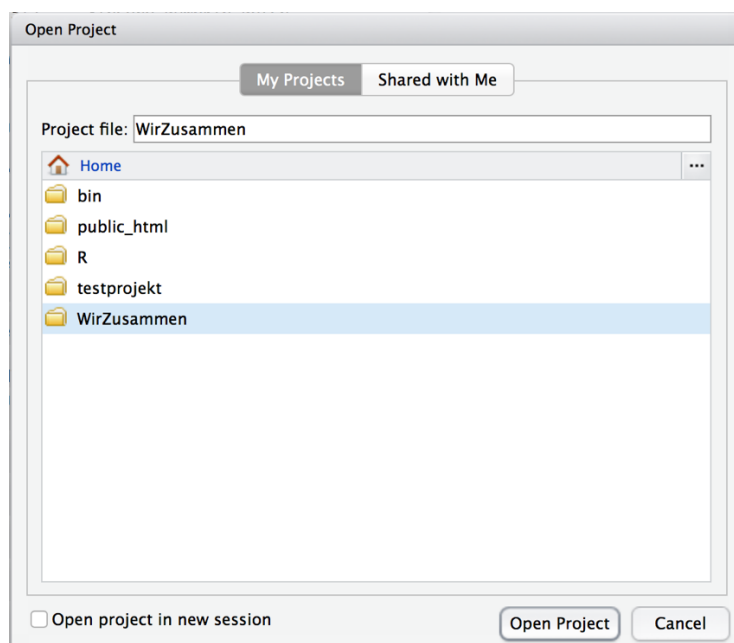
Der R-Server übernimmt hierbei das Management der Dateien, so daß tatsächlich mehrere Benutzer gleichzeitig mit der Originaldatei arbeiten können. Die Benutzer müssen sich auch um nichts kümmern, weil der R-Server die gesamte Dateiverwaltung übernimmt.

## Projekt freigeben

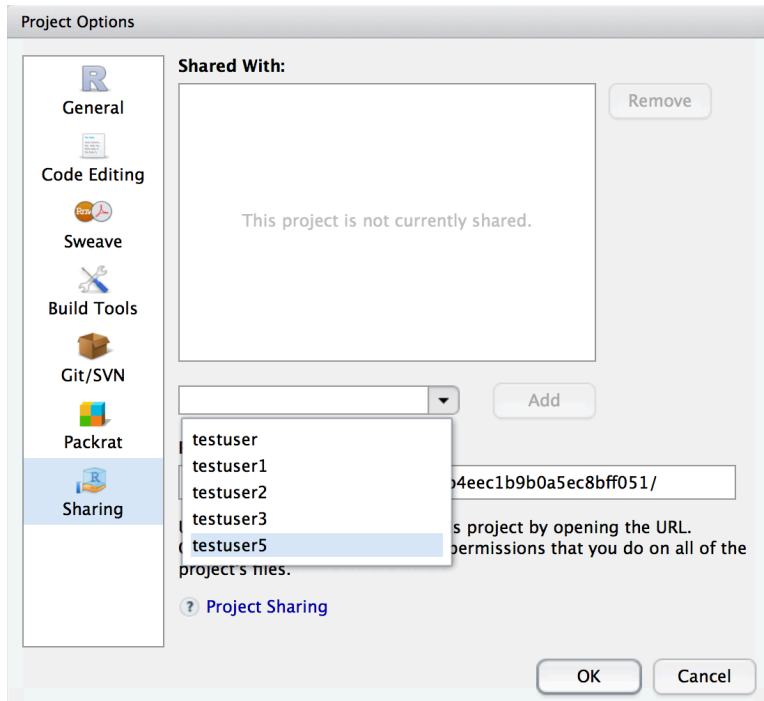
Im folgenden Beispiel möchte der Benutzer ein Projekt mit dem Namen *WirZusammen* freigeben.



Zunächst ist das gewünschte Projekt auszuwählen, indem in der rechten oberen Ecke mit einem Klick auf den Pfeil hinter *Project* das Menü aufgerufen wird.



Das gewünschte Projekt wird ausgewählt und auf *Open Project* geklickt.



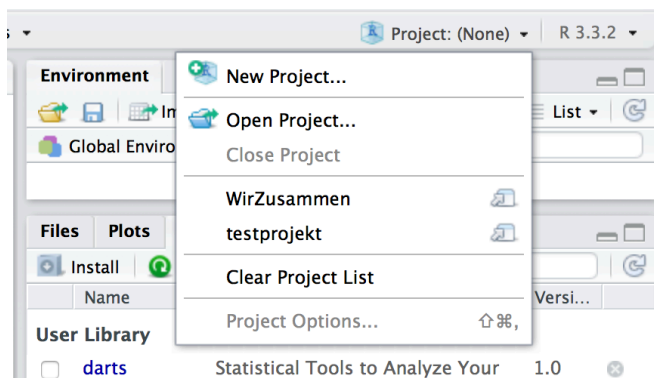
Nun kann in dem Pull-Down-Menü in der Mitte der Benutzer ausgewählt werden, dem die Zusammenarbeit gestattet werden soll. Hier wird der *testuser5* ausgewählt und mit *Add* hinzugefügt.

Unten wird die URL angezeigt, mit der der R-Server das Projekt verwaltet. Den Pfad zu den Originaldateien bekommen die Teilnehmer nicht zu sehen.

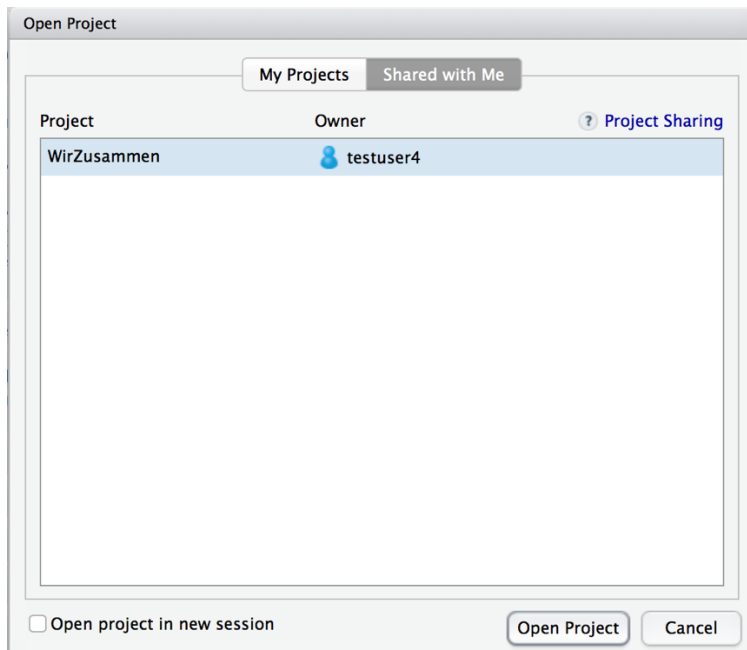
*Nur der ursprüngliche Projektstarter sollte andere Benutzer zu Teilnehmern erklären, damit es kein Chaos gibt. Die Teilnehmer können sich auch gegenseitig hinauswerfen; lediglich den Projektstarter kann niemand löschen.*

## Projekt als Teilnehmer öffnen

Im folgenden Beispiel möchte der Benutzer das Projekt mit dem Namen *WirZusammen* als Teilnehmer öffnen.

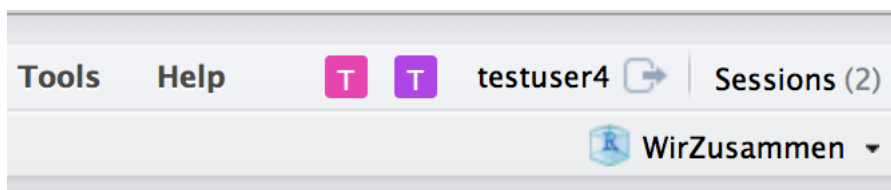


Zunächst ist das gewünschte Projekt auszuwählen, indem in der rechten oberen Ecke mit einem Klick auf den Pfeil hinter *Project* das Menü aufgerufen wird. Dann wird *Open Project* ausgewählt.

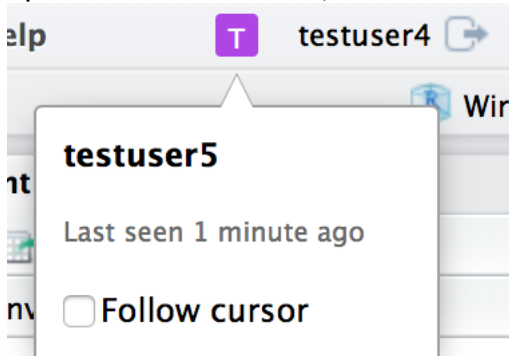


Der Tab *Shared with Me* zeigt alle Projekte, zu denen der Benutzer als Teilnehmer zugelassen wurde. Der Benutzer wählt das Projekt und klickt auf *Open Project*.

### Aktive Teilnehmer erkennen




Sowie ein anderer Benutzer gleichzeitig an diesem Projekt angemeldet ist, erscheint ein Symbol oben im Fenster; verschiedene Farben stehen hier für verschiedene Benutzer.



Bewegt man den Mauszeiger auf das Symbol, so erscheint ein kleines Fenster, in dem man sehen kann, woran der andere Teilnehmer gerade arbeitet und wann er zuletzt aktiv war.

```
1 Diesen Text schrieb User 4
2 #Hey, du pappnase: Kommentare benötigen ein #!
3 print("Hello World")xx
```

Änderungen erscheinen sofort bei allen Teilnehmern. Hier stammt die erste Zeile von User 4, die mittlere hat User 5 beigesteuert und die Funktion kommt von User 2.

Eine besondere Bedeutung kommt dem Kästchen *Follow Cursor* zu. Wird diese angeklickt, so verändert sich das kleine Symbol . Ab diesem Zeitpunkt wird dann die Cursorposition des entsprechenden Teilnehmers für den eigenen Cursor übernommen.

*Diese Funktion ist sehr praktisch, wenn man sich etwas in einem längeren Script zeigen lassen möchte, da man sofort sieht, wo der andere Teilnehmer Änderungen vornimmt.*

*Besonders hilfreich ist es, wenn die Teilnehmer untereinander Sprechkontakt haben, z.B. per Telefon.*

## Ressourcenverwaltung

Im Unterschied zu HLR und HLR64 übernimmt der R-Server die Verwaltung und Zuteilung der Ressourcen (Prozessoren, Speicher, Rechenzeit) selbst.

Alle Benutzer sind derzeit gleichberechtigt, es gibt lediglich die Ausnahme, daß die RAM-Verwendung auf 8 GB bei Kursteilnehmern und 16 GB bei anderen Benutzern limitiert ist. Diese Begrenzung gibt es nur deshalb, damit ein evtl. Programmierfehler nicht unbeabsichtigt zu einem Totalausfall des Systems führen kann.

Verschiedene Stellschrauben ermöglichen eine Optimierung des Systems, wobei es sich aber stets nur um relative Veränderungen handelt, deren Auswirkungen bisher nicht genau bekannt sind.

*Es wird zunächst davon ausgegangen, daß die Entwickler schon wissen, was sie tun, und daher eine sinnvolle Voreinstellung abliefern.*

Sollten sich die Voreinstellungen zu irgendeinem Zeitpunkt als problematisch herausstellen, so kann im Gespräch mit dem Administrator versucht werden, eine Lösungsmöglichkeit zu erarbeiten.

*Beispielsweise benötigt ein Benutzer für eine sehr umfangreiche Matrixberechnung die kompletten 128 GB RAM; Lösung: am Wochenende, außerhalb der Kurszeiten sollte das möglich sein.*

## R-Server und lokales R

Es ist problemlos möglich, daß der R-Server von einem Client genutzt wird, auf dem ein lokales R installiert ist. Die Systeme beeinflussen sich nicht gegenseitig und könnten sogar zeitgleich genutzt werden. Eine gemeinsame Benutzung von Paketen ist jedoch nicht möglich.

## von R-Studio empfohlene Browserversionen

- Mozilla Firefox ab Version 10
- Apple Safari ab Version 5.1
- Google Chrome ab Version 21
- Microsoft Internet Explorer ab Version 10

Bei den folgenden Versionen traten Probleme auf

- IE 6/WindowsXP
- IE 9/Windows7
- Firefox 14/SUSE Linux 12.2