Leibniz Supercomputing Centre
of the Bavarian Academy of Sciences and Humanities

Using R at LRZ

April 2021

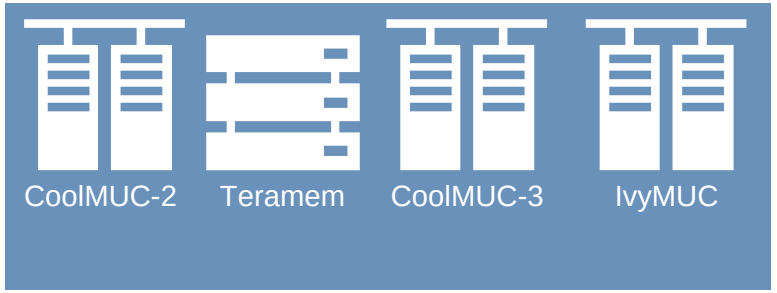
Course Information

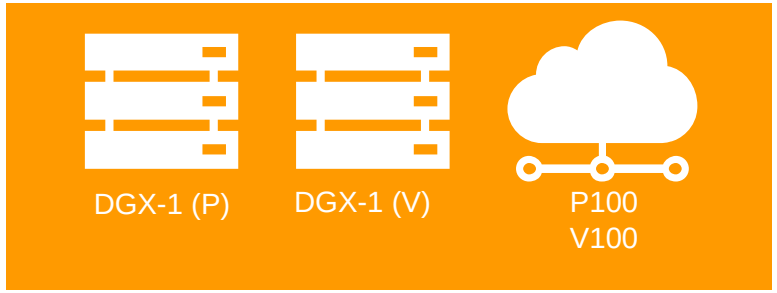
- The aim of this course is to demonstrate the different ways of using R efficiently and productively on LRZ systems (with some focus on machine learning tasks)
- It is not an introduction to R itself
- Many of the topics covered in this course are based on issues encountered by users, for which they created tickets at the LRZ Servicedesk
- Also, it assumes you have some prior knowledge and experience in using GNU/Linux and SSH (if you attended Tuesday's courses, you should be fine)

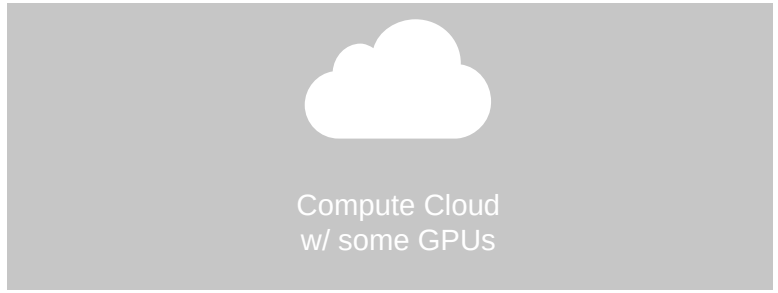
HPC Systems for Bavarian Universities

DSS
(Data Science Storage)

Tape Archive
and Backup







lxlogin8.lrz.de

lxlogin[1-4].lrz.de lxlogin10.lrz.de datalab2.srv.lrz.de https://cc.lrz.de

https://www.rstudio.lrz.de

https://www.rstudio.lrz.de

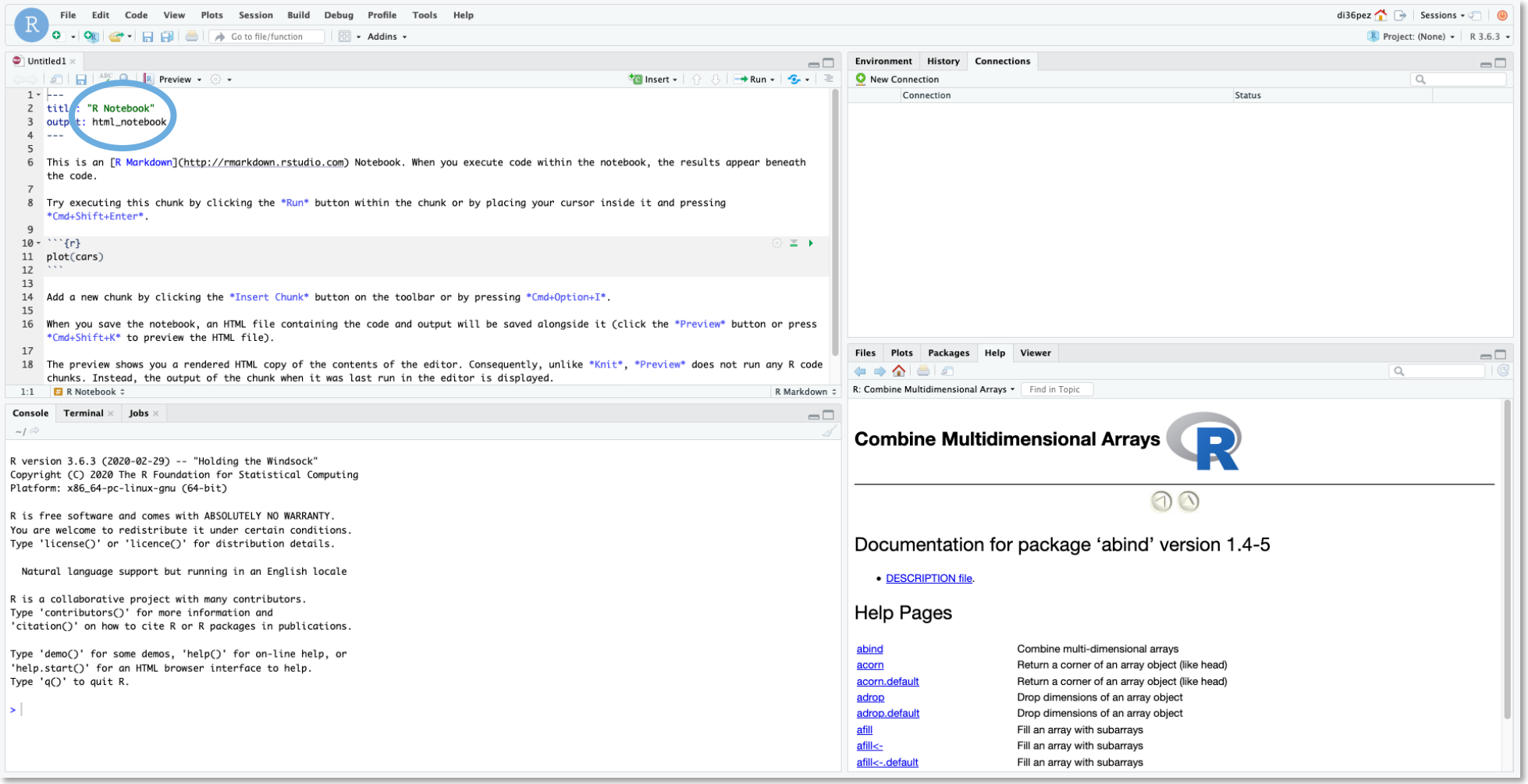
RStudio Server

- Web-based RStudio frontend
- Cluster of multiple nodes, with
  - 40 cores and
  - 360 GB RAM each
- Integrates with the Linux Cluster:
  - Directly access the data in your DSS-backed Linux Cluster home directory ($HOME)
  - Allows to access any DSS-based storage container
    (NFS-Export has to be set up by data curator)
  - Use the built-in Terminal to submit jobs to the Linux Cluster's batch queues via the Slurm Workload Manager

- For further details, see https://doku.lrz.de/x/zQWVAg

https://www.rstudio.lrz.de

RStudio Server



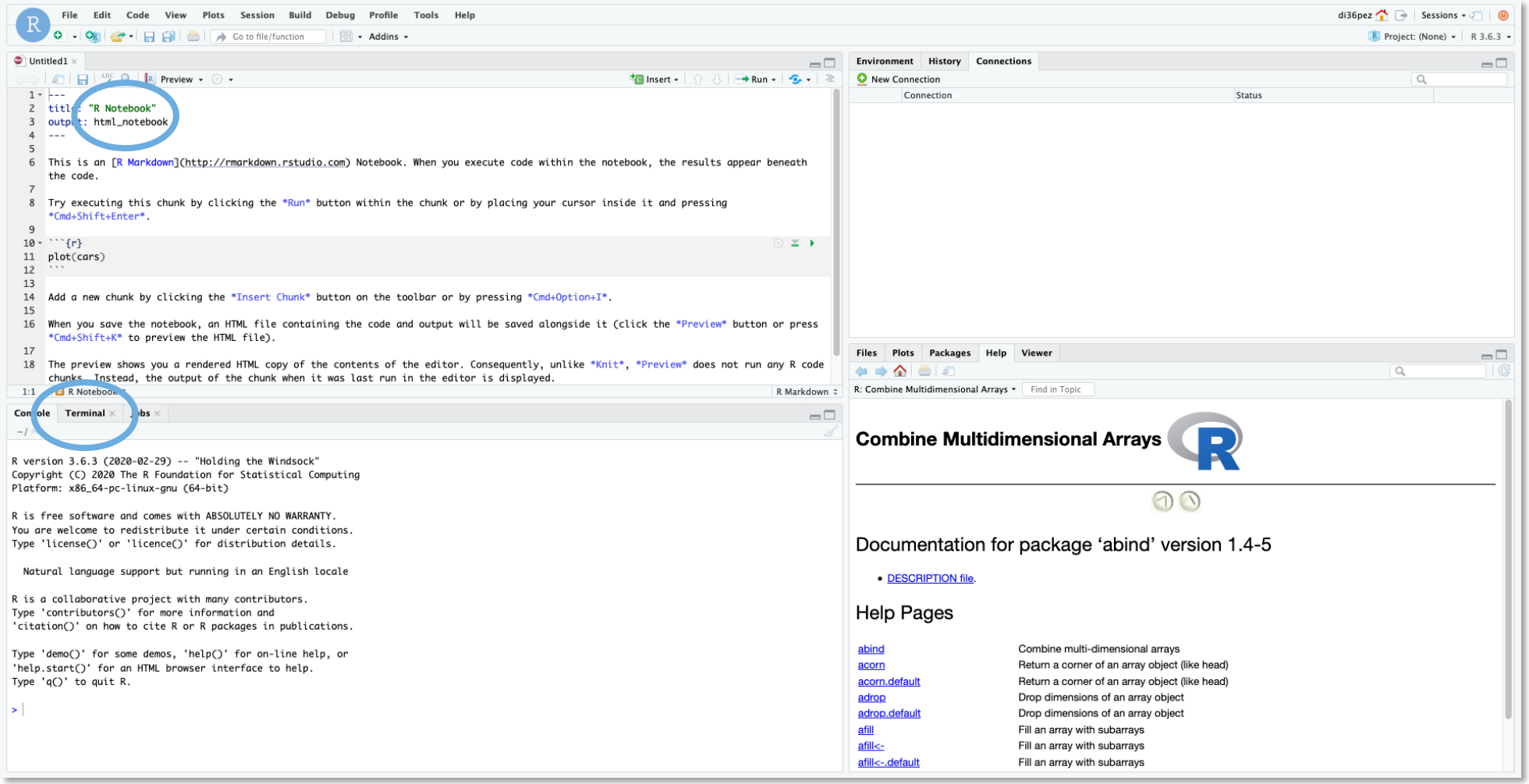
https://www.rstudio.lrz.de

RStudio Server

- R Notebooks:
  R Markdown documents with code chunks that can be executed independently and interactively, with output visible immediately beneath the input

https://www.rstudio.lrz.de

RStudio Server



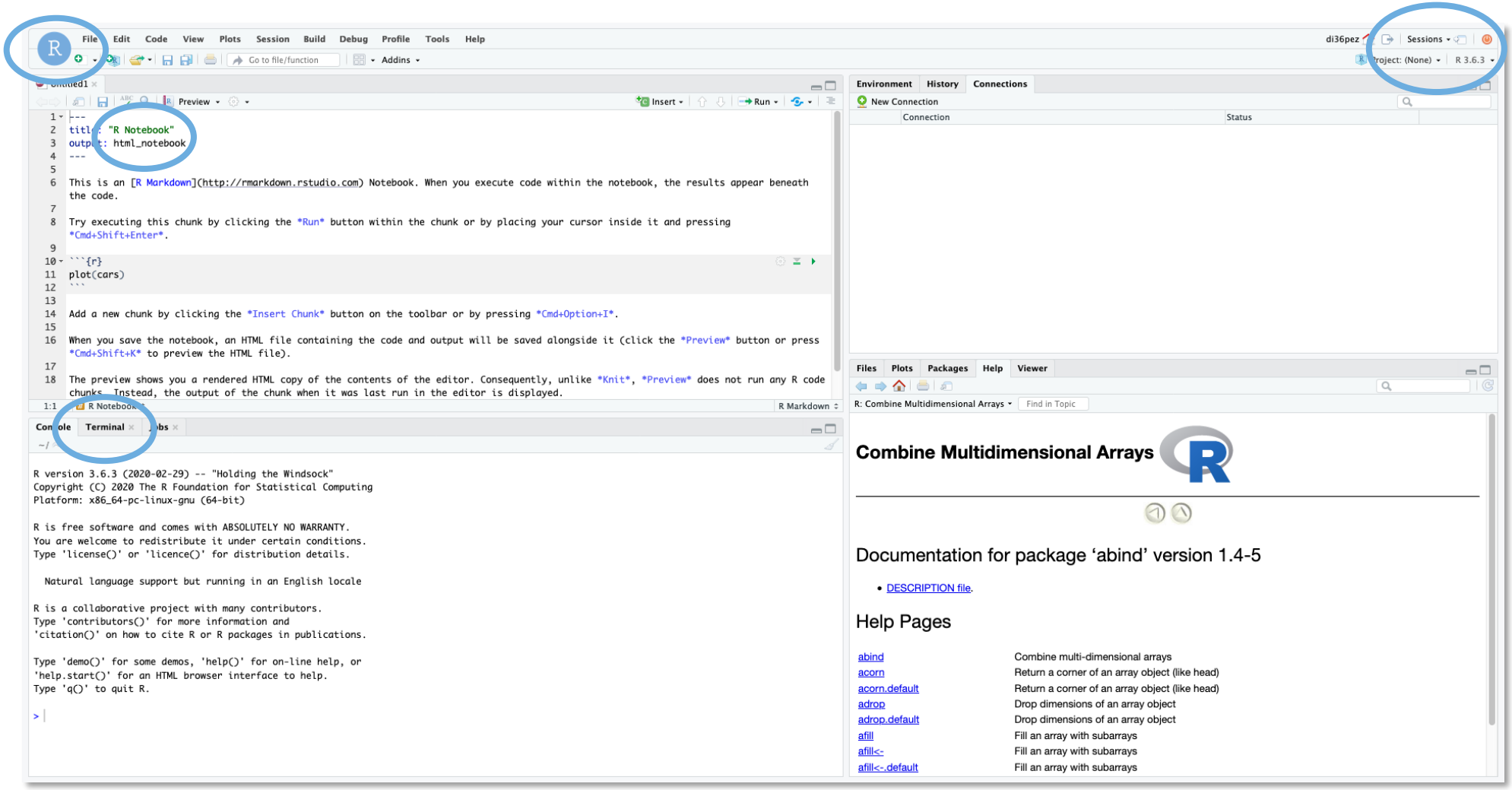
https://www.rstudio.lrz.de

RStudio Server

- Integrated Terminal:
  Provides access to the system shell from within Rstudio
- Can be used to submit jobs to the Slurm workload manager of CoolMUC-2

https://www.rstudio.lrz.de

RStudio Server



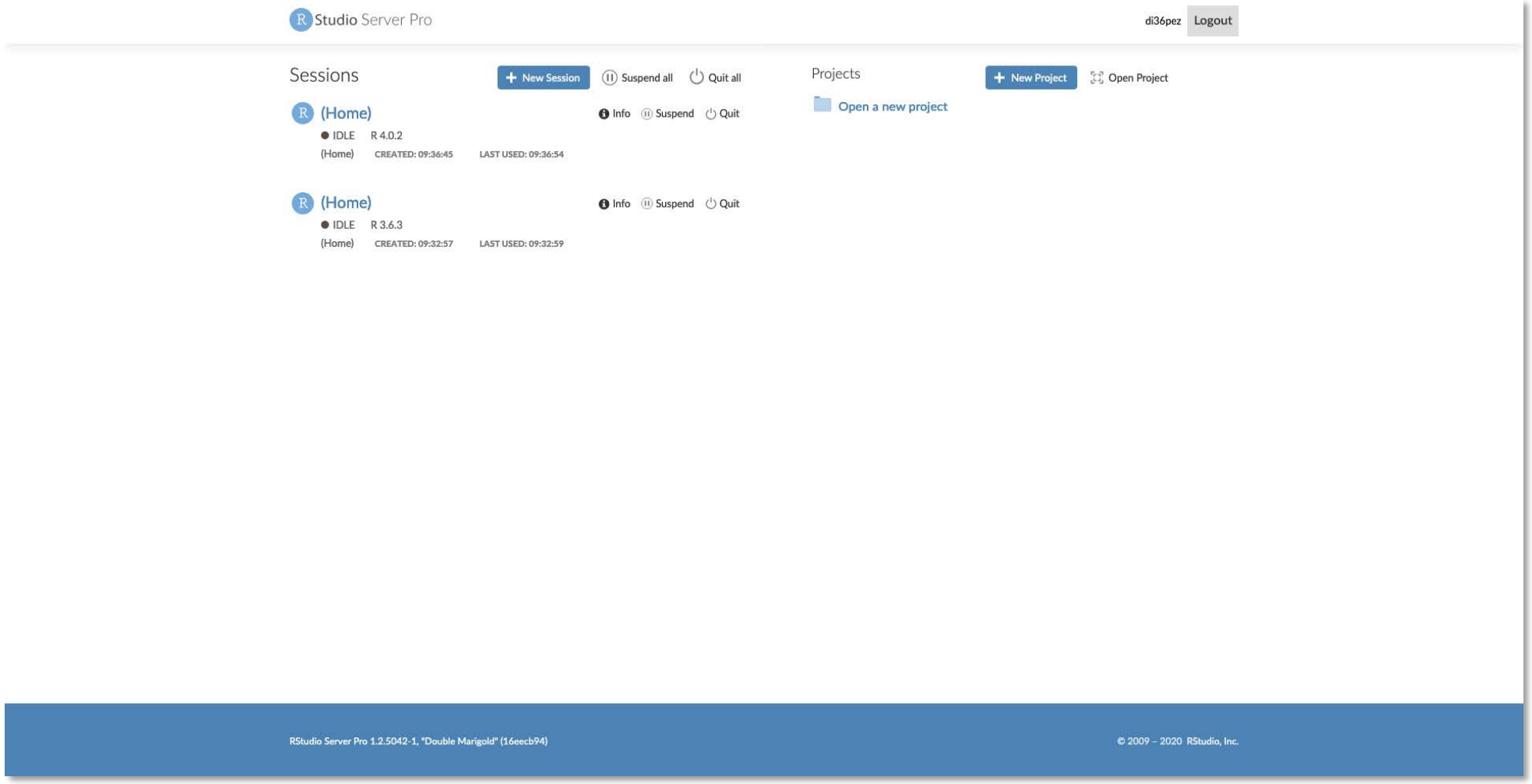
https://www.rstudio.lrz.de

RStudio Server

- You can open multiple concurrent sessions
  (please don‘t use more than 5 at any given time!)
- This can be used to run multiple analyses in parallel (even using different versions of R) and they can be kept open (almost) indefinitely

https://www.rstudio.lrz.de

RStudio Server



Linux Cluster

- Connect to the CoolMUC-2 segment of the Linux Cluster
- From a terminal application:
  `$ ssh <user>@lxlogin1.lrz.de`
- Alternatives would be lxlogin[2-4].lrz.de for CoolMUC-2 or lxlogin8.lrz.de for CoolMUC-3 or lxlogin10.lrz.de for IvyMUC

R Modules

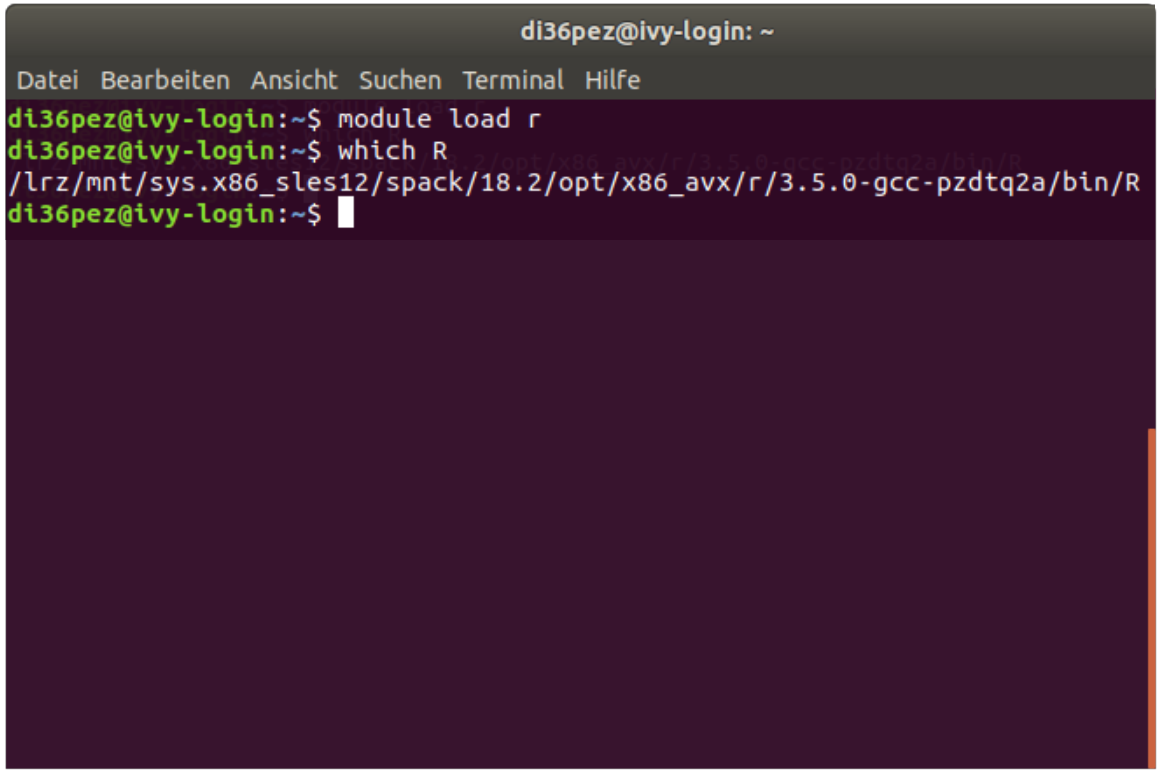
- R is not accessible on the Linux Cluster by default (try: `$ which R`)
- Environment modules allow for the dynamic modification of environment variables
- A (minimal) set of default modules is active after login:
  `$ module list`
- Use the module system to search for different R versions:
  `$ module available r` (or `module av r`)

R Modules

```
di36pez@ivy-login: ~
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe
di36pez@ivy-login:~$ which R
which: no R in (/lrz/sys/intel/studio2017_u6/impi/2017.4.256/lrzbin:/lrz/sys/int
el/studio2017_u6/impi/2017.4.256/bin64:/lrz/sys/intel/studio2017_u6/compilers_an
d_libraries_2017.6.256/linux/bin/intel64:/lrz/sys/share/modules/bin:/lrz/sys/bin
:/usr/local/bin:/usr/bin:/bin:/usr/bin/X11:/usr/games:/opt/ibutils/bin:/lrz/sys/
tools/slurm_utils/bin)
di36pez@ivy-login:~$ module list
Currently Loaded Modulefiles:
 1) admin/1.0     3) intel/17.0   5) mpi.intel/2017       7) lrz/default
 2) tempdir/1.0   4) mkl/2017     6) spack/release/18.2
di36pez@ivy-login:~$ module av r
-------------------- /lrz/sys/share/modules/files/graphics ---------------------
rvsvnc/1.0(default)
-------------------- /lrz/sys/share/modules/files/libraries --------------------
root/6.12(default)
---------------------- /lrz/sys/share/modules/files/tools ----------------------
redis/3.2.5(default)
----------- /lrz/sys/spack/18.2/modules/x86_avx/linux-sles12-x86_64 ------------
r/3.4.4-X11       r/3.5.0-X11       readline/7.0
r/3.4.4-X11-mkl   r/3.5.0-X11-mkl   renderproto/0.11.1
di36pez@ivy-login:~$
```

R Modules

- (The current default version of) R can be loaded using
  `$ module load r`
- If you need a different version, you have to specify the full name of the module, e.g. “r/3.4.4-gcc8-mkl”



R Modules

- We are using the package manager Spack (https://spack.io) to provide applications/modules
- Spack “meta modules” make the (additional) module path(s) available
- By default, the latest LRZ release of Spack is loaded (cf. `$ module list`)
- Going forward, there might be newer (pre-release) versions of the Spack software stack available (e.g. spack/staging/20.2, spack/develop) which might then also provide newer versions of R
- If in doubt, stick to the final releases (i.e. spack/release/YY.X)!

R Package Management

- All R packages are installed into libraries – these are (just) directories in the file system with subdirectories for each installed package
- The default installation of R comes with a single library (R_HOME/library) usually containing the standard and recommended packages
  (in RStudio, this is called the System Library)
- On a multiuser system, regular users may not add/install packages directly into this library (but administrators can)
- For the latest versions of R on the Linux Cluster we only provide the standard set of base packages in this central location

R Package Management

- Individual users can have (one or more) additional, personal libraries (called User Library in RStudio)
- The path for this library directory can be specified by the environment variable $R_LIBS_USER (amongst others)
- If this is not defined, R will ask you to create a personal package library when installing packages for the first time…

R Package Management

```
di36pez@ivy-login: ~
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe
R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

> install.packages("ggplot2")
Warnung in install.packages("ggplot2")
  'lib = "/lrz/mnt/sys.x86_sles12/spack/18.2/opt/x86_avx/r/3.5.0-gcc-pzdtq2a/rli
b/R/library" ist nicht schreibbar
Would you like to use a personal library instead? (yes/No/cancel) yes
Would you like to create a personal library
‘~/R/x86_64-pc-linux-gnu-library/3.5’
to install packages into? (yes/No/cancel)
```

- Notice the suggested path – it is specific to the (minor) version of R!
- You can use the .libPaths() function within R to check the current library directories…

R Package Management

```
di36pez@ivy-login: ~
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe
di36pez@ivy-login:~$ R

R version 3.5.0 (2018-04-23) -- "Joy in Playing"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

> .libPaths()
[1] "/home/hpc/pr28fa/di36pez/R/x86_64-pc-linux-gnu-library/3.5"
[2] "/lrz/mnt/sys.x86_sles12/spack/18.2/opt/x86_avx/r/3.5.0-gcc-pzdtq2a/rlib/R/l
ibrary"
>
```

- So, subject to the system/cluster segment and R version you‘re using, you will depend on different system and user libraries
- You can always control the R packages you use (and their versions) by maintaining your user library…
- … it might be beneficial to do this in a project-specific manner.

R Package Management

- The challenge: on GNU/Linux (most) „add-on“ R packages will be compiled from source
- This requires compilers, tools and additional dependencies available on the system
- For general compatibility use (a recent version) of the GNU Compiler Collection GCC to compile add-on packages:

```
module unload intel-mpi
module unload intel
module load gcc
module load r
```

- If you miss any dependencies, make sure to check the available modules!
- And, as always: if you encounter any problems, please talk to us!

R Package Management

- Optional:
  there are package managers which can be run as user applications and may provide additional dependency requirements
- They manage R and (many of) its packages „from the outside“
- You could take a look at Spack (https://spack.io), conda (https://conda.io) or Homebrew (https://brew.sh)

Slurm Workload Manager

- Slurm is a job scheduler:
  - Allocates access to resources (time, memory, nodes/cores)
  - Provides framework for starting, executing, and monitoring work
  - Manages queue of pending jobs (enforcing “fair share” policy)
- Use the `sinfo` command to get information about the available clusters
  `$ sinfo --clusters=all` or, shortened:
  `$ sinfo –M all`

Slurm Workload Manager

```
di36pez@mpp2-login5: ~
Datei  Bearbeiten  Ansicht  Suchen  Terminal  Hilfe
di36pez@mpp2-login5:~$ sinfo -M all
CLUSTER: bsbslurm
PARTITION     AVAIL  TIMELIMIT  NODES  STATE NODELIST
bsb_konvert*     up   infinite      1    mix hbsbr09c05s02
bsb_konvert*     up   infinite      1  alloc hbsbr09c05s01
bsb_konvert*     up   infinite      4   idle hbsbr09c05s[03-06]

CLUSTER: hm_mech
PARTITION       AVAIL  TIMELIMIT  NODES  STATE NODELIST
hm_mech_batch*     up 14-00:00:0     12  alloc hhmkr09c04s[01-12]

CLUSTER: httf
PARTITION    AVAIL  TIMELIMIT  NODES  STATE NODELIST
httf_batch*     up 3-00:00:00      5   resv httfr05c05s[01-05]

CLUSTER: htus
PARTITION    AVAIL  TIMELIMIT  NODES  STATE NODELIST
htus_batch*     up 3-00:00:00      2   idle htusr05c04s[05-06]

CLUSTER: inter
PARTITION      AVAIL  TIMELIMIT  NODES  STATE NODELIST
mpp3_inter*       up    2:00:00      1  alloc mpp3r03c05s03
mpp3_inter*       up    2:00:00      2   idle mpp3r03c05s[01-02]
teramem_inter     up 4-00:00:00      1    mix teramem1
```

- Look for the cluster segments
  - inter (allows for interactive usage)
  - cm2 (the main CoolMUC-2 cluster)
  - serial (shared nodes for serial jobs)
- What is their current status?
- Get information about a specific cluster segment, e.g.
  `$ sinfo -M inter` or
  `$ sinfo -M cm2`

CoolMUC-2 Overview

Slurm Cluster	Slurm Partition	Node Range	Slurm Job Settings

cm2	cm2_large	25-64	`--clusters=cm2`
			`--partition=cm2_large`
			`--qos=cm2_large`

	cm2_std	3-24	`--clusters=cm2`
			`--partition=cm2_std`
			`--qos=cm2_std`

cm2_tiny	cm2_tiny	1-4	`--clusters=cm2_tiny`

serial	serial_std	1	`--clusters=serial`
			`--partition=serial_std`
			`--mem=<memory_per_node>MB`

	serial_long	1	`--clusters=serial`
			`--partition=serial_long`
			`--mem=<memory_per_node>MB`

inter	cm2_inter	1-4	`--clusters=inter` `--partition=cm2_inter`
	teramem_inter	1	`--clusters=inter` `--partition=teramem_inter`

For additional details see https://doku.lrz.de/display/PUBLIC/Job+Processing+on+the+Linux-Cluster

Interactive R Session

- The inter cluster can be used for interactive resource allocation:
  `$ salloc -p cm2_inter -n 1`
- Using this shell, you can e.g. run R interactively on this node (if the R module is loaded):
  `$ R`

Interactive R Session

```
user@cm2login1:~$ salloc -p cm2_inter -n 1
salloc: Granted job allocation 159945
user@i22r07c05s11:~$ module load r
user@i22r07c05s11:~$ R

R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

[...]

> library(parallel)
> detectCores()
[1] 56
>
```

Job Processing

- For production jobs, you want to prepare and submit batch scripts
- They tell Slurm about the resources you need and the scripts/programs you want to run…

Job Processing

```
#!/bin/bash
#SBATCH --clusters=cm2_tiny
#SBATCH --nodes=1

module load slurm_setup

module load r

Rscript myscript.R
```

- A very minimal example of a job script (not necessarily recommended, but working in some cases), requesting
  - a single, exclusive node (with 28 cores)
  - of the cm2_tiny partition/cluster, part of
  - the CoolMUC-2 system
- Submit this job script to the queue:
  $ sbatch <myjob.sh>

Job Processing

```
#!/bin/bash
#SBATCH -o /dss/dsshome1/.../.../myjob.%j.%N.out
#SBATCH -D /dss/dsshome1/.../.../workdir
#SBATCH -J jobname
#SBATCH --get-user-env
#SBATCH --clusters=cm2
#SBATCH --partition=cm2_std
#SBATCH --nodes=3
#SBATCH --mail-type=end
#SBATCH --mail-user=xyz@xyz.de
#SBATCH --export=NONE
#SBATCH --time=08:00:00

module load slurm_setup

module load r
cd workdir

mpirun R –f myscript.R
```

- A more practical example…
  - defining custom output file(s)
  - setting a working directory
  - assigning a job name
  - configuring mail notifications
  - managing the environment
  - limiting walltime explicitly
- See documentation for more details: https://doku.lrz.de/x/AgaVAg

Job Management and Accounting

- Submit a job:
  ```
  $ sbatch myjob.sh
  ```
- Query status of your jobs:
  ```
  $ squeue -M mpp2 -u <user>
  ```
- Approximate start time of pending jobs:
  ```
  $ squeue -M mpp2 -u <user> --start
  ```
- Abort a job:
  ```
  $ scancel -M mpp2 <jobid>
  ```
- Get accounting data for (past) jobs:
  ```
  $ sacct -X -M mpp2 [-S <YYYY-MM-DD>] -u <user>
  ```

Potential Pitfalls

- Jobs get aborted (by Slurm) if they use more resources than specified
  -> you need to estimate memory and runtime requirements
  - Estimate memory requirements from a (single, local) serial run, extrapolate if needed (use e.g. your system monitor or the “top” command line tool)
  - Provide some “buffer” for runtime
- Queuing times can be long
  - Use “sinfo” to find less busy cluster segments
  - Smaller, less demanding jobs generally start faster
    -> you can benefit from accurate resource estimation

Potential Pitfalls

- Debugging can be inconvenient
- The time interval between changes in the R code and seeing results/getting feedback is longer than usual
- The compute environment (compute nodes of the cluster) and the development/test environments (local, login or interactive nodes) are usually not exactly the same
  - Debug as much as possible in a serial fashion
  - Prepare small jobs and test them interactively (using “salloc”)