Your 3D environment (aka: scene) normally has several elements and those elements each have their own properties. One element of particular importance is ?you?, the viewer of the scene. If you aren?t in a room, you can?t be expected to see what is in that room. With 3D scenes, the viewer is usually referred to as the camera. (Unlike in 2D where it?s often called a window or view)

[...]

Part of the camera?s properties are the near and far clipping planes, which specify the closest point to the camera which is visible, and the furthest away point. Anything closer than the near plane, or further away than the far plane, will be clipped and hence invisible.

Of course, you can get something in between. If your cube is 200 units across, sitting at 900 units from the camera, and the far plane is at 1000 units ? you will see half of it.

The solution here is to set the near and far plane distances appropriately to the scene you?re working in: sometimes this is easy, everything is a similar scale and stays a consistent distance to the camera. Other times, it?s a huge topic which requires redesigning your renderer to avoid artefacts : especially when you have large distances or tiny objects. For more on this, and why selecting good near/far value is hard, read up ?depth buffer precision?.

Limesurvey is an online survey tool. It is very powerful and commonly used in academic environments because it is Free Software (GPLv2+), allows for local installations protecting the data of participants and allowing to comply with data protection regulations. This also
means there are typically no load-balanced multi-server szenarios with HA databases. But simple VMs where Limesurvey runs and needs upgrading in place.

There's an LTS branch (currently 3.x) and a stable branch (currently 4.x). There's also a 2.06 LTS branch that is restricted to paying customers. The main developers behind Limesurvey offer many services from template design to custom development to support to hosting ("Cloud", "Limesurvey Pro"). Unfortunately they also charge for easy updates called "ComfortUpdate" (currently 39€ for three months) and the manual process is made a bit cumbersome to make the "ComfortUpdate" offer more attractive.

- **RcppGSL 0.3.8: More fixes and polish** [4]

  Release 0.3.8 of RcppGSL is now getting onto CRAN. The RcppGSL package provides an interface from R to the GNU GSL using the Rcpp package.

  Peter Carbonetto let us know in issue #25 that the included example now showed linker errors on (everybody?ts favourite CRAN platform) Slowlaris. Kidding aside, the added compiler variety really has benefits because we were indeed missing a good handful or two of inline statements in the headers?which our good friends g++ and clang++ apparently let us get away with. This has been fixed, and a little bit of the usual package polish and cleanup has been added; see the list of detailed changes below.

- **LLVM Is Looking At Establishing An "Incubator" Process For Encouraging New Sub-Projects** [5]

  In addition to changing the acceptable language within the LLVM project, another topic this week sure to be interesting is on the establishing of an "incubator" process similar to that of Apache Incubator projects.

  Due to the rather high bar currently set for accepting new LLVM sub-projects, LLVM project founder Chris Lattner has proposed an "incubator" process after the idea was presented by one of his former colleagues.

- **T^4 #7 and R^4 #5: R and CRAN Binaries for Ubuntu** [6]

  A new video in both our T^4 series of video lightning talks with tips, tricks, tools, and toys is also a video in the R^4 series as it revisits a topic previously covered in the latter: how to (more easily) get (binary) packages onto your Ubuntu system. In fact, we show it in three different ways.