GG606

Misc

Reminder

- *Tri-Agency Research Data Management Policy*
  - http://www.science.gc.ca/eic/site/063.nsf/eng/h_97610.html
  - Applies to institutions & researchers
  - Data management plans
- *Institutional research data management strategies*
  - https://science.gc.ca/site/science/en/interagency-research-funding/policies-and-guidelines/research-data-management/published-institutional-research-data-management-strategies
  - https://www.wlu.ca/academics/research/research-services/assets/resources/research-data-management-institutional-strategy.html
  - https://uwaterloo.ca/research/research-data-management-institutional-strategy

Deliverable 1 follow-up

- Superscripts & subscripts not easy
- See `demo(plotmath)` for examples
  - Similar to LaTeX
  - `bquote()` or `paste()` or `expression()`
  - Sometimes depends on your `device`

Deliverable 1 follow-up

- Superscripts & subscripts not easy
- ```
  qplot(1,1) +
  labs(x = expression("NO"[3]^-{}*" (mgN/L)"),
  y = expression("Q (m"^2*"/s)"))
  ```
- `[3]` subscript
- `^2` superscript, need `{}` after minus since there is no number
- `*` to paste things together
  ```

Spatial Data

- `sp` & `sf` packages
  - https://cran.r-project.org/web/views/Spatial.html
  - `sf` articles https://r-spatial.github.io/sf/
  - `sp` example https://edzer.github.io/sp/
  - Significant OS-specific dependencies (rdgal, gdal, geos, rwinlib, lwgeom)
- Data Carpentry has 3 workshops
  (last one is best)
  - https://datacarpentry.org/geospatial-workshop/

Model Fitting

- R4DS: Chapters 22-25
- `purrr::map` act on nested data.frames
- `broom` turn tidy models into tidy data
- Hard to generalise

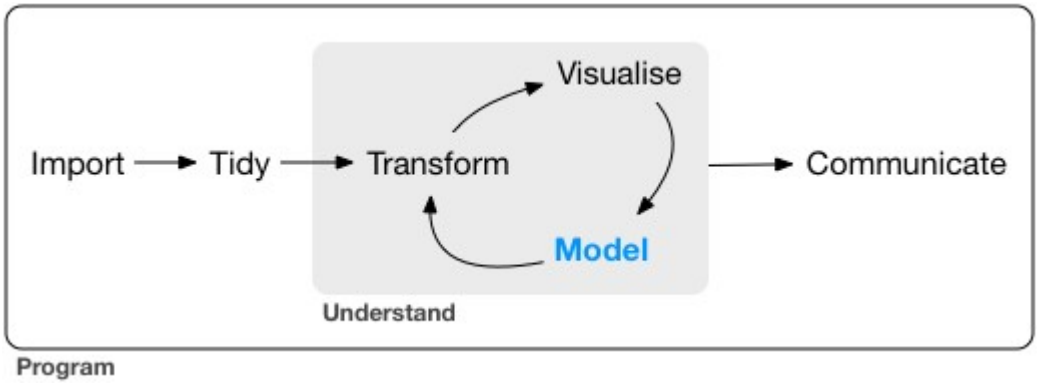
Model

22 **Introduction**

23 Model basics

24 Model building

25 Many models



Model Fitting

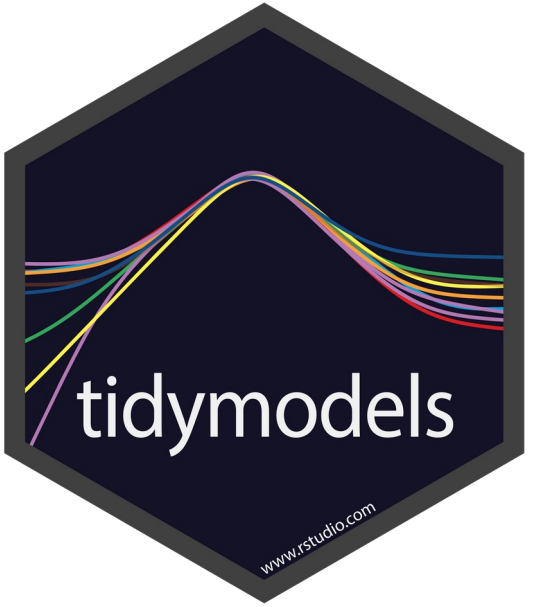
- `lm()` & compare AIC values?
- Generalised linear `glm()` or generalised additive `mgcv::gam()` or penalised linear `glmnet::glmnet()` or Robust linear `MASS::rlm()`
- Trees `rpart::rpart()` or `randomForest::randomForest()`

Model Fitting|Building

- Differential Equations
  - https://cran.r-project.org/web/views/DifferentialEquations.html
- `deSolve` for coupled differential equations & `simecol` (+FME) for simulations & fitting
- Asks different question than statistical model

Tidymodels

- https://www.tidymodels.org/
- Sampling, unified interface, workflows, performance tuning



Shiny

- https://shiny.rstudio.com/tutorial/
- Videos & lessons
- Hard to so much code
- Similar to html and css

Functions, Loops

- `for (i in X){}`
- `for (i in seq_along(X)){}`
- `apply(array, margin, fun)`
  - `apply, lapply, sapply, vapply`
- Mostly preference

Loops

- `for (i in X){}`
- Use vectorised equivalent if it exists
- Don’t grow objects in loop (e.g. `c`, `cbind`, `rbind`)
- Prealloc object(s) & fill during loop
- `*apply` handles memory alloc
- `purrr:map*` functions more consistent than `*apply`

Loops

- Don’t grow objects in loop (e.g. `c`, `cbind`, `rbind`)
- Store outputs as a list:
  ```
  out ← vector("list",
  length(inputs))
  ```
- `unlist(out)` or
  `purrr::flatten_dbl(out)`

Loops

- `means ← c(0, 1, 2)`
- ```
  output ← double()
  for (i in seq_along(means)) {
    n ← sample(100, 1)
    output ← c(output, rnorm(n, means[[i]]))
  }
  ```
- `str(output)`
  ```

Loops

- `out ← vector("list", length(means))`
- ```
  for (i in seq_along(means)) {
    n ← sample(100, 1)
    out[[i]] ← rnorm(n, means[[i]])
  }
  ```
- `str(out)`
- `unlist(out)`
- `purrr::flatten_dbl(out)`
  ```

Conditional ‘Loops’

- Simpler than for loop bc it only has 2 components
- Best for simulations?
- Special types of iterations when total number of iterations is not known

```
while (condition) {
  # body
}
```

Next

- Presentations: March 21 & 28
  - *Written:*
  - Technical depth – /40
  - Critique – /25
  - Accuracy – /20
  - Writing style – /15
  - *Presentation:*
  - Aesthetic appeal – /25
  - Clarity and communication style – /25
  - Technical completeness – /50
- The report should be no longer than 4000 words
- The presentation should be between 13-15 minutes