Review of R and RStudio

EES 5891-03

Bayesian Statistical Methods

Jonathan Gilligan

Class #5: Thursday, September 08 2022

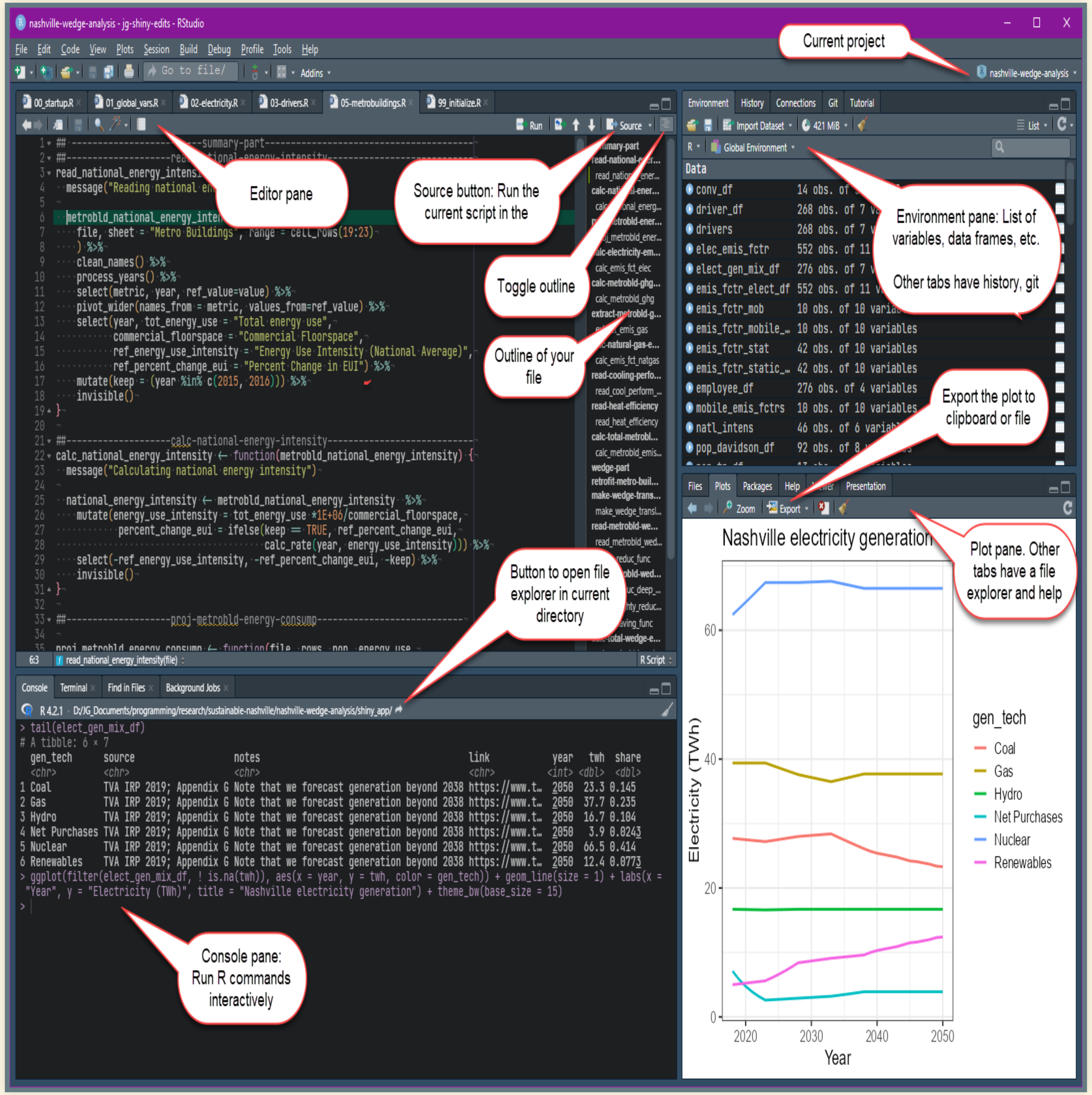
Announcement

- I have updated the instructions for configuring R and downloading necessary packages (in Homework #1 and on the “Tools” page of the web site).
- The `rethinking` package now uses `cmdstanr` as its default for Monte Carlo sampling , instead of `rstan`. `cmdstanr` is more reliable and up to date; `rstan` is being phased out.

Overview of Rstudio

RStudio Window layout

- Project oriented
  - A project is a directory with a group of related files (R, data, etc.)
  - A project can be managed with `git` for revision control
  - Create a new project:
    - Create a new directory for a project
    - Create a project in an existing directory
    - Download a project from GitHub or another external source, using `git`
- Online help and cheatsheets
- Window layout
  - Four panes:
    1. Editor pane (edit R scripts, text files, etc.)
    2. Console and Terminal pane (interactively run R commands, etc.)
    3. Files, Plots, Packages, Help, etc.
       - You can export plots to the clipboard or files.
    4. Environment, History, Git, etc.



nashville-wedge-analysis - jg-shiny-edits - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/
Addins
nashville-wedge-analysis
00_startup.R
01_global_vars.R
02-electricity.R
03-drivers.R
05-metrobuildings.R
99_initialize.R
Run
Source
Environment History Connections Git Tutorial
Import Dataset
421 MiB
List
R
Global Environment
Console Terminal Find in Files Background Jobs
Files Plots Packages Help Viewer Presentation
Zoom
Export
6:3
read_national_energy_intensity(file)
R Script
R 4.2.1 · D:/JG_Documents/programming/research/sustainable-nashville/nashville-wedge-analysis/shiny_app/
Nashville electricity generation
Electricity (TWh)
Year
0
20
40
60
2020
2030
2040
2050
gen_tech
Coal
Gas
Hydro
Net Purchases
Nuclear
Renewables

Current project

Editor pane

Source button: Run the current script in the

Toggle outline

Outline of your file

Environment pane: List of variables, data frames, etc.

Other tabs have history, git

Export the plot to clipboard or file

Plot pane. Other tabs have a file explorer and help

Button to open file explorer in current directory

Console pane: Run R commands interactively

```r
## ---------------------summary-part----------------------------------------
##------------------rea          tional-energy-intensity--------------------
read_national_energy_intensi
  message("Reading national en

  metrobld_national_energy_inten
    file, sheet = "Metro Buildings", range = cell_rows(19:23)
    ) %>%
    clean_names() %>%
    process_years() %>%
    select(metric, year, ref_value=value) %>%
    pivot_wider(names_from = metric, values_from=ref_value) %>%
    select(year, tot_energy_use = "Total energy use",
           commercial_floorspace = "Commercial Floorspace",
           ref_energy_use_intensity = "Energy Use Intensity (National Average)",
           ref_percent_change_eui = "Percent Change in EUI") %>%
    mutate(keep = (year %in% c(2015, 2016))) %>%
    invisible()
}

##------------------calc-national-energy-intensity--------------------------
calc_national_energy_intensity ← function(metrobld_national_energy_intensity) {
  message("Calculating national energy intensity")

  national_energy_intensity ← metrobld_national_energy_intensity  %>%
    mutate(energy_use_intensity = tot_energy_use *1E+06/commercial_floorspace,
           percent_change_eui = ifelse(keep == TRUE, ref_percent_change_eui,
                                       calc_rate(year, energy_use_intensity))) %>%
    select(-ref_energy_use_intensity, -ref_percent_change_eui, -keep) %>%
    invisible()
}

##------------------proj-metrobld-energy-consump----------------------------

proj_metrobld_energy_consump ← function(file, rows, pop, energy_use,
```

```
Data
conv_df            14 obs. of
driver_df          268 obs. of 7 v
drivers            268 obs. of 7 v
elec_emis_fctr     552 obs. of 11
elect_gen_mix_df   276 obs. of 7 v
emis_fctr_elect_df 552 obs. of 11 v
emis_fctr_mob      10 obs. of 10 varia
emis_fctr_mobile_… 10 obs. of 10 variables
emis_fctr_stat     42 obs. of 10 variables
emis_fctr_static_… 42 obs. of 10 variables
employee_df        276 obs. of 4 variables
mobile_emis_fctrs  10 obs. of 10 variables
natl_intens        46 obs. of 6 variab
pop_davidson_df    92 obs. of 8
```

```
> tail(elect_gen_mix_df)
# A tibble: 6 × 7
  gen_tech      source                  notes                                 link          year   twh  share
  <chr>         <chr>                   <chr>                                 <chr>        <int> <dbl>  <dbl>
1 Coal          TVA IRP 2019; Appendix G Note that we forecast generation beyond 2038 https://www.t…  2050  23.3 0.145
2 Gas           TVA IRP 2019; Appendix G Note that we forecast generation beyond 2038 https://www.t…  2050  37.7 0.235
3 Hydro         TVA IRP 2019; Appendix G Note that we forecast generation beyond 2038 https://www.t…  2050  16.7 0.104
4 Net Purchases TVA IRP 2019; Appendix G Note that we forecast generation beyond 2038 https://www.t…  2050   3.9 0.0243
5 Nuclear       TVA IRP 2019; Appendix G Note that we forecast generation beyond 2038 https://www.t…  2050  66.5 0.414
6 Renewables    TVA IRP 2019; Appendix G Note that we forecast generation beyond 2038 https://www.t…  2050  12.4 0.0773
> ggplot(filter(elect_gen_mix_df, ! is.na(twh)), aes(x = year, y = twh, color = gen_tech)) + geom_line(size = 1) + labs(x =
 "Year", y = "Electricity (TWh)", title = "Nashville electricity generation") + theme_bw(base_size = 15)
> 
```

R Language

R Language

- R is flexible
- Two approaches
  - Base R
    - Very flexible and powerful
    - Sometimes confusing and verbose
    - Nothing special about data framse (`data.frame`)
  - Tidyverse (`library(tidyverse)`)
    - Unified philosophy of data analysis
      - Canonical reference: R for Data Science
      - Oriented toward `data.frames` (and `tibbles`)
      - Principles of “Tidy Data”
      - Consistent approach makes it easy to figure out how to do what you want to do.
      - Download extensive cheatsheets via RStudio help menu.

Graphics

- Base R
  - plot command

```
plot(x, y) # plot y vs. x with points
plot(x, y, type = "l") # plot with a
line
plot(height ~ weight, data = df) # with
a data.frame
```

- Tidyverse/ggplot2 (library(tidyverse) or library(ggplot2))
  - ggplot command

```
data(Howell1)
ggplot(Howell1, aes(x = weight, y = height)) +
  geom_point() +
  labs(x = "weight (kg)", y = "height (cm)",
    title = "!Kung San height and weight")
ggplot(Howell1, aes(x = weight, y = height)) +
  geom_point(aes(shape = male, size = age),
             "color = "magenta") +
  labs(x = "weight (kg)", y = "height (cm)",
    title = "!Kung San height and weight")
```

  - Philosophy: Grammar of Graphics
    - online manual https://ggplot2-book.org/
    - data, mapping (aesthetics), layers (geometries), scales, coordinates, facets, and themes
    - You can combine and adjust these different parts separately.

Data

- Base R
  - Separate 1-dimensional lists or vectors of data
  - 2D (or higher) arrays: `data.frame`, `array`, `matrix`
  - Index rows and columns `Howell1[107, 3]`
    - `Howell1[107,]` for all columns of the row
    - `Howell1[,"height"]` for all rows of the "height" column
    - `Howell1[c(1,3,5),10:15]` to get rows 1, 3, and 5 of columns 10–15.
    - `Howell1["age" >= 18, c("height", "weight")]`

- Tidyverse (`library(tidyverse)`)
  - `data.frame`
  - `tibble` (an enhanced `data.frame`)
  - Select columns:

```
select(Howell1, height, weight, age)
select(Howell1, -male, -age)
```

    - Select all columns that start with "`foo_`" but don't end with "`bar`"

```
select(my_data, starts_with("foo_"),
       -ends_with("bar"))
```

  - Select rows:

```
filter(Howell1, age >= 18, male)
```

Tidyverse Data Manipulations

- Modifying data: `mutate`

```
mutate(Howell1, hgt_std = (height - mean(height)) /
sd(height),
                wt_std = (weight - mean(weight)) /
sd(weight))
```

- Summarizing data

```
summarize(d, height = mean(height), weight =
mean(weight))
d_tmp <- group_by(d, male)
d_tmp <- summarize(d_tmp, height = mean(height),
                   weight = mean(weight))
d_tmp <- ungroup(d_tmp)
```

- Pivot tables: `pivot_longer`, `pivot_wider`
  - Example using `relig_income` data

Pipe operator

- It can get confusing to combine multiple commands

```
ungroup(summarize(group_by(filter(Howell1, age >= 18), male),
  height = mean(height), weight = mean(weight)))
```

- Pipe commands allow us to break this up:

```
Howell1 %>% filter(age >= 18) %>%
  group_by(male) %>%
  summarize(height = mean(height), weight = mean(weight)) %>%
  ungroup()
```

- The pipe operator `%>%` sends it input (what's on the left) to the first argument of the function on the right.
  - `Howell1 %>% filter(age >= 18)` is the same as `filter(Howell1, age >= 18)`
  - `Howell1 %>% filter(age >= 18) %>% group_by(male)` is the same as `group_by(filter(Howell1, age >= 18), male)`

Sampling from models

- Base R

```
mdl <- quap(alist(
  height ~ dnorm(mu, sigma),
  mu <- a + b * weight,
  a ~ dnorm(178,20),
  b ~ dlnorm(0,1)
  sigma ~ dunif(0,50)
  ), data = d2)
w_lst <- data.frame(weight = seq(30, 70, by =
5))
```

  - Sample from posterior predictive distribution for data (e.g., `height`):

```
extract.samples(mdl, 1000)
```

  - Sample from posterior of model link (`mu`):

```
link(mdl, w_list)
```

- Tidyverse (`library(tidyverse)`)

```
library(tidybayes)
library(tidybayes.rethinking)
```

  - Sample from posterior predictive distribution for data (e.g., `height`):

```
predicted_draws(w_lst, mdl, ndraws = 1000,
                value = "height")
add_predicted_draws(mdl, w_lst, ndraws = 1000,
                    value = "height")
```

  - Sample from posterior of model link (`mu`):

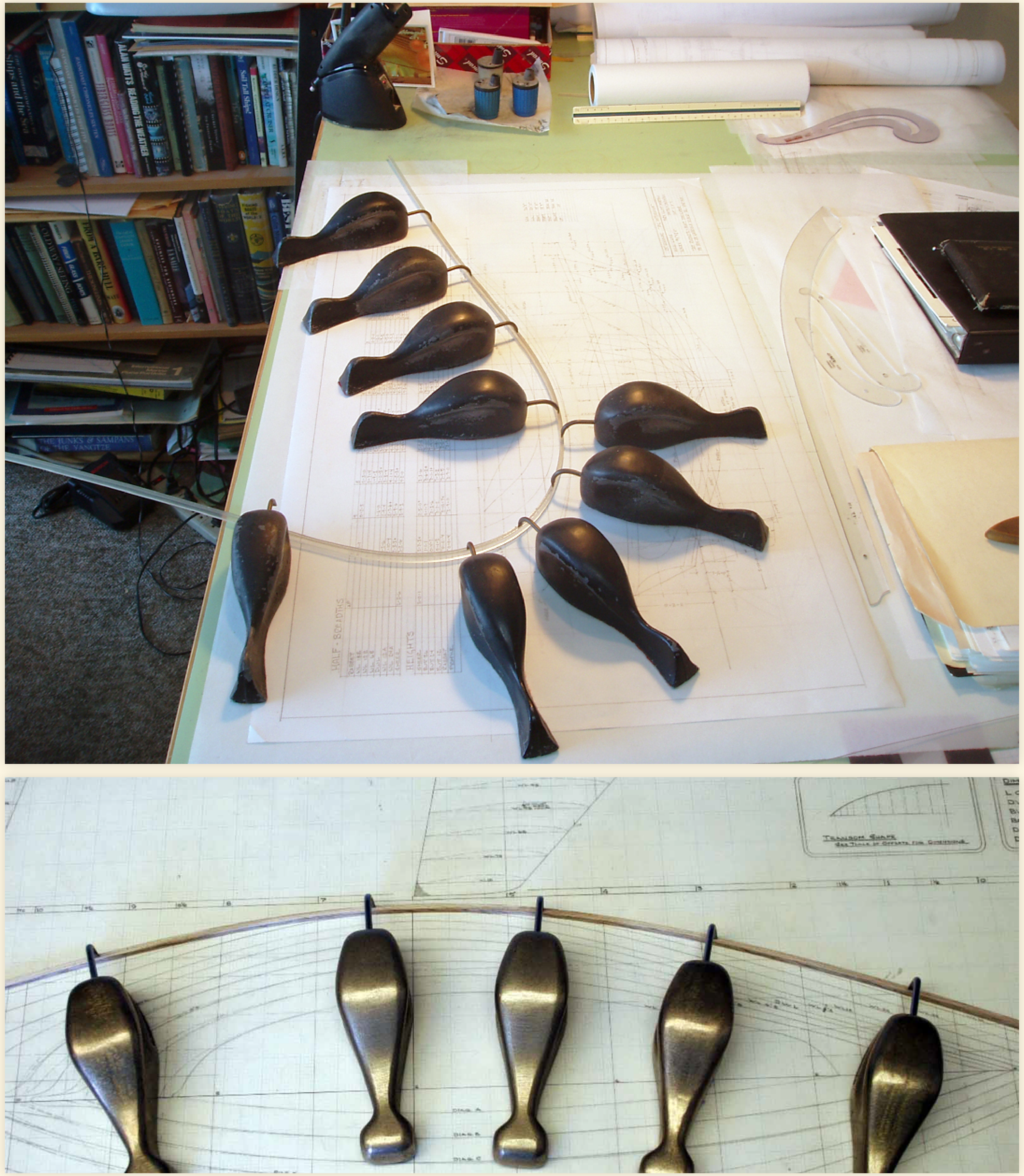
```
linpred_draws(w_lst, mdl, ndraws = 1000,
              value = "height")
add_linpred_draws(mdl, w_lst, ndraws = 1000,
                  value = "height")
```

- `add_` versions are the same, but reverse the order of the first two arguments (model object and new data).

Splines

Splines

- Originally from mechanical drafting splines
- Arbitrary smooth curve
- Complexity:
  - Physical splines: “ducks” or “whales”
  - Mathematical splines: “knots”
- Splines are a special case of a class of models called *generalized additive models* (GAMs).



(Photos: Rain Noe, When Splines Were Physical Objects)

Splines in Statistical Regression

- Cherry blossom data
  - Over 1000 years of historical records for timing of full-blossoming of cherry trees
    - Y. Aono & S. Saito, Int. J. Biometeorology **54**, 211 (2010).
  - No changes for most of history, but pronounced trend in 20th century (global warming).
- Spline regression:
  - *Basis splines*: for the ith point, x_i

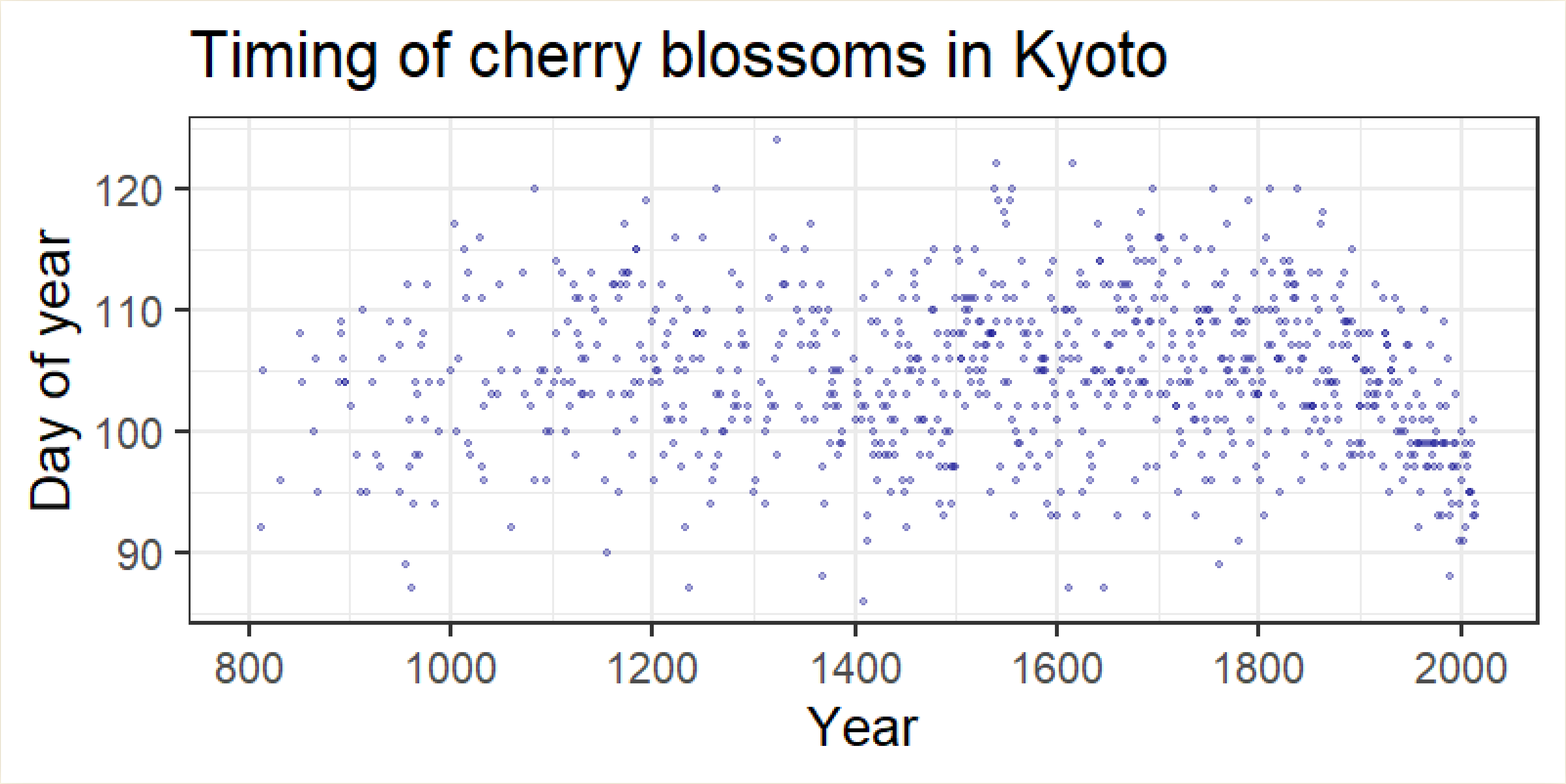
$$\mu_i = \alpha + \sum_{j=1}^{n_{\text{knots}}} w_j B_{i,j}$$

n_{knots} = # knots, w_j = weight for knot j, $B_{i,j}$ = ith row of jth basis function (matrix with one row for each x value, and n_{knots} columns).

```
data(cherry_blossoms)
d <- cherry_blossoms
precis_show(precis(d, digits = 2))
```

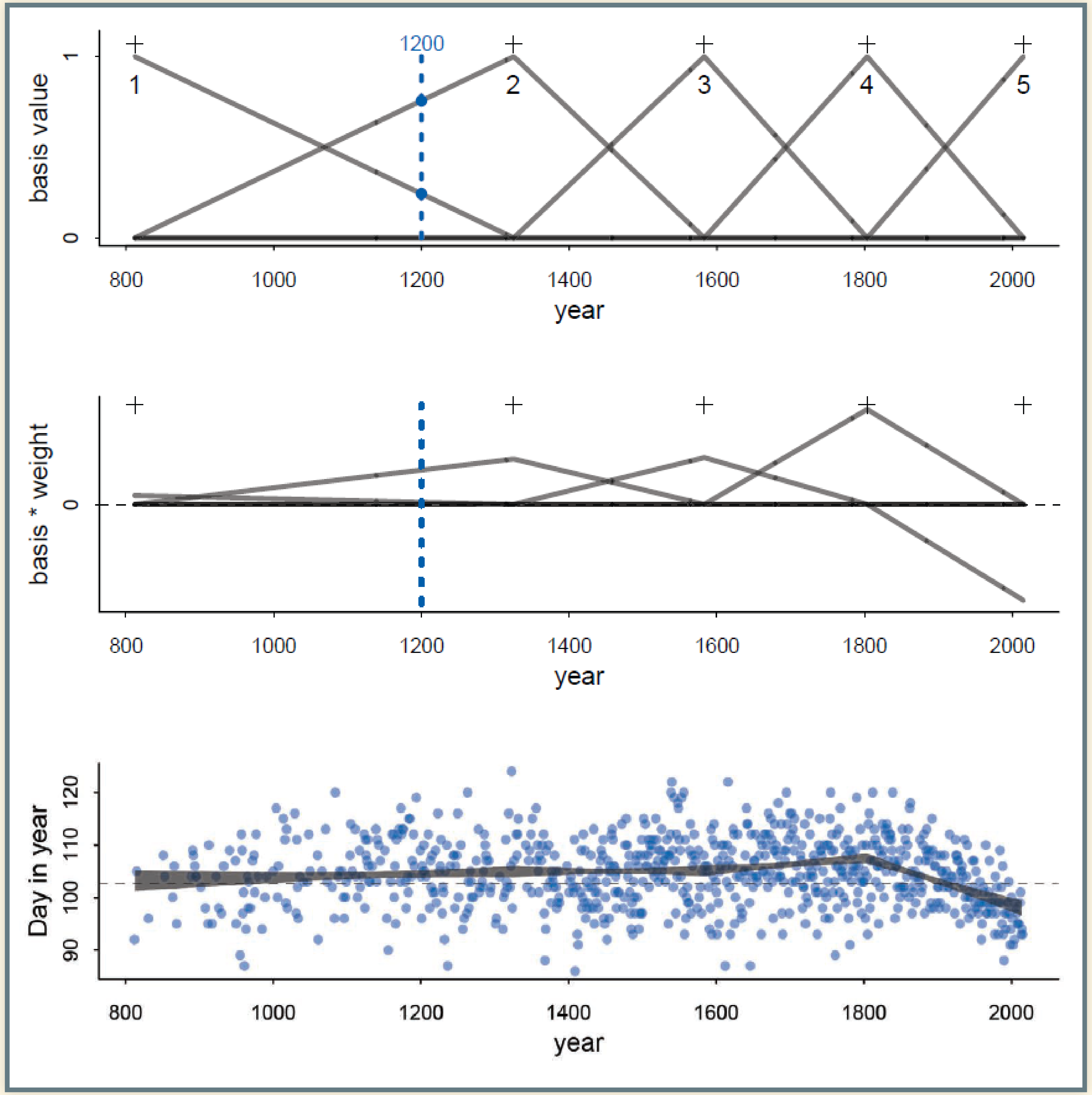
```
## 'data.frame': 1215 obs. of 5 variables:
##                 mean     sd    5.5%   94.5%       histogram
## year         1408.00 350.88  867.77 1948.23
## doy           104.54   6.41   94.43  115.00
## temp            6.14   0.66    5.15    7.29
## temp_upper      7.19   0.99    5.90    8.90
## temp_lower      5.10   0.85    3.79    6.37
```

```
ggplot(d, aes(x = year, y = doy)) +
  geom_point(color = "darkblue", size = 1, alpha = 0.3) +
  scale_x_continuous(breaks = seq(600, 2200, by = 200)) +
  labs(x = "Year", y = "Day of year",
       title="Timing of cherry blossoms in Kyoto")
```



Linear Basis Spline

- Linear basis functions $B_j(x)$
  - 5 knots
  - Piecewise linear
  - At most 2 functions are nonzero for any x.
- Model fits weights w_j for each basis function



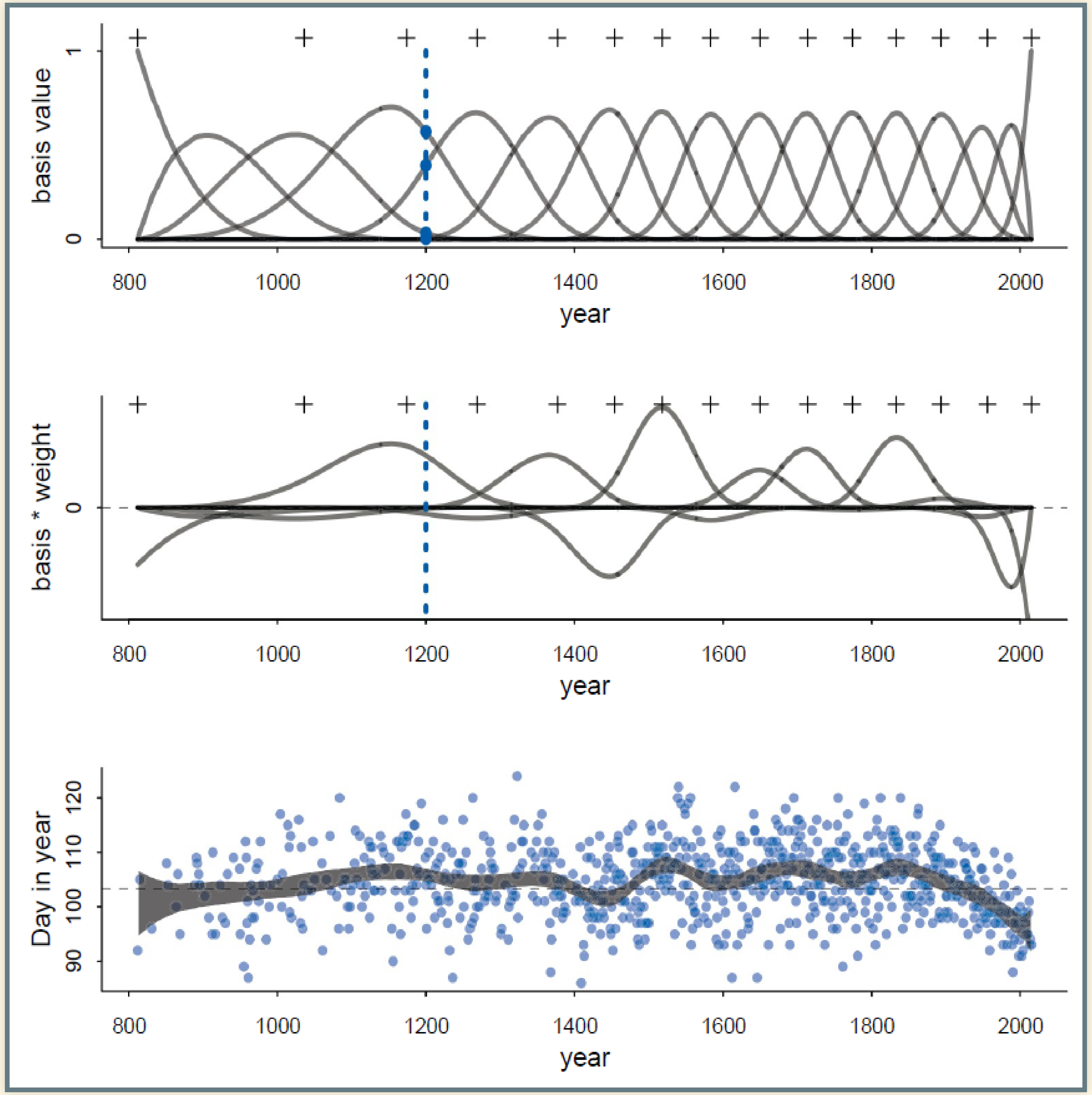
Cubic Basis Spline

- 15 knots
  - Equal # of years with data between knots.
- Cubic functions
- Only 3 have nonzero values for any *x*.

```
library(splines)
d2 <- filter(d, ! is.na(doy)) # omit missing values
n_knots <- 15
knot_list <- quantile(d2$year,
                      probs=seq(0,1, length.out =
        n_knots))

# Create basis function matrix
B <- bs(d2$year, knots = knot_list[-c(1,n_knots)],
        degree = 3, intercept = TRUE)

mdl <- quap(
  alist(
    D ~ dnorm(mu, sigma),
    mu <- a + B %*% w, # %*% is matrix multiplication
    a ~ dnorm(100, 10),
    w ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ),
  data = list(D = d2$doy, B = B),
  start = list(w = rep(0, ncol(B)))
)
```


basis value
year
basis * weight
year
Day in year
year