Geocentric Models

EES 4891-06/5891-01

Bayesian Statistical Methods

Jonathan Magnolia Gilligan

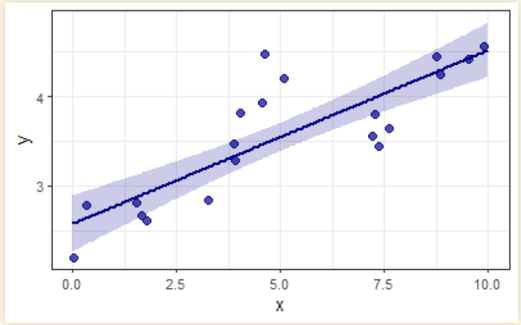
Class #5: Wednesday, January 21 2026

Big Picture

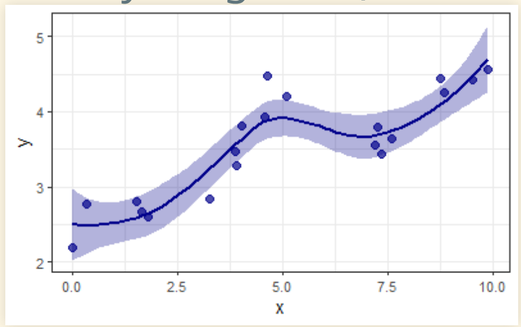
Characteristics of Models

- Parametric regression models
  - Prediction: $y = f(x)$, where f is characterized by one or more *parameters*
    - Linear regression ($y = \alpha + \beta x$)
    - Polynomial regression ($y = \alpha + \sum_{i=1}^{N} \beta_i x^i$)
    - Generalized linear models ($y = f(\alpha + \beta x)$)
    - Nonlinear models
- Nonparametric regression models
  - y can't be described as a simple function of x:
    - Spline models
    - Locally-weighted regression
    - Gaussian process models
    - …
- **Comparison**
  - Which fits the data best?
  - Which would be best for interpolating?
  - Which would be best for extrapolating?

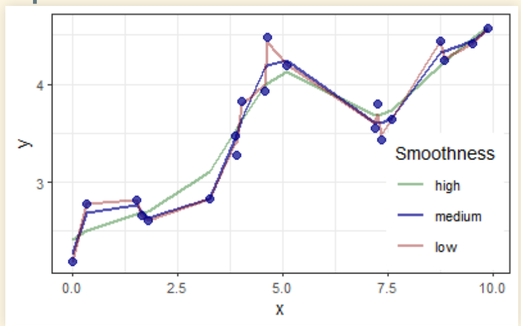
- Parametric models:
  - Linear



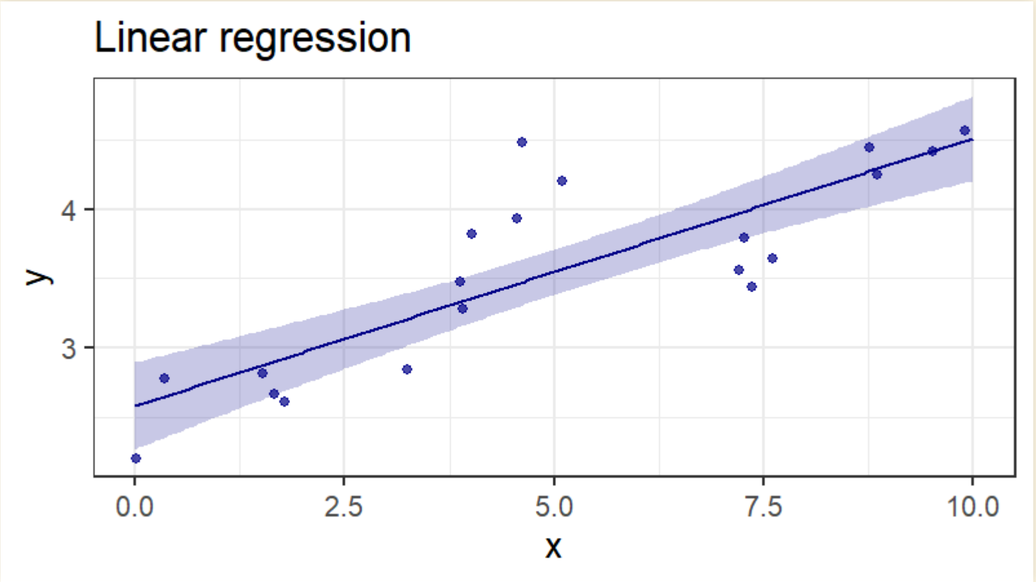
- Nonparametric
  - Locally weighted (LOWESS)

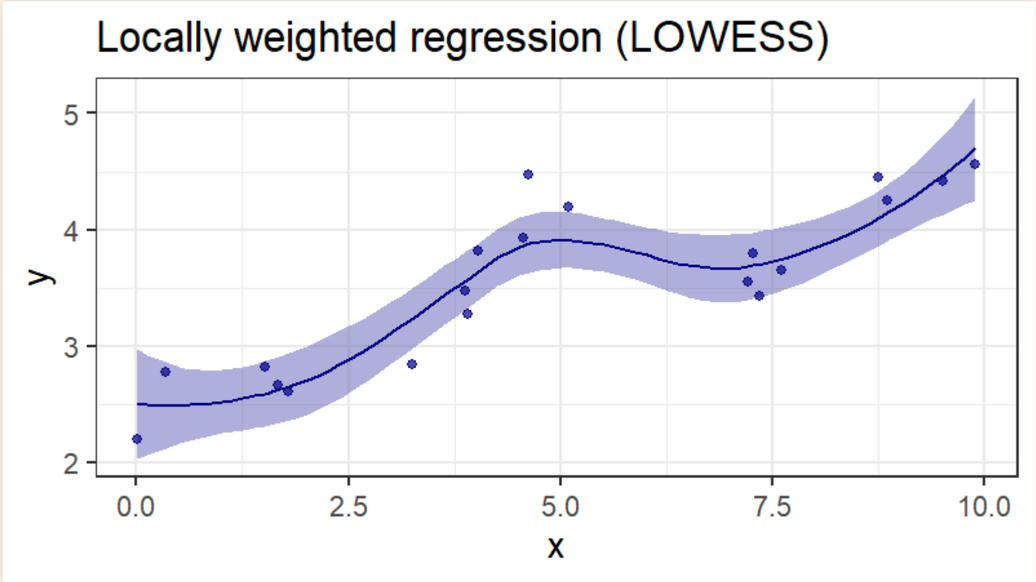


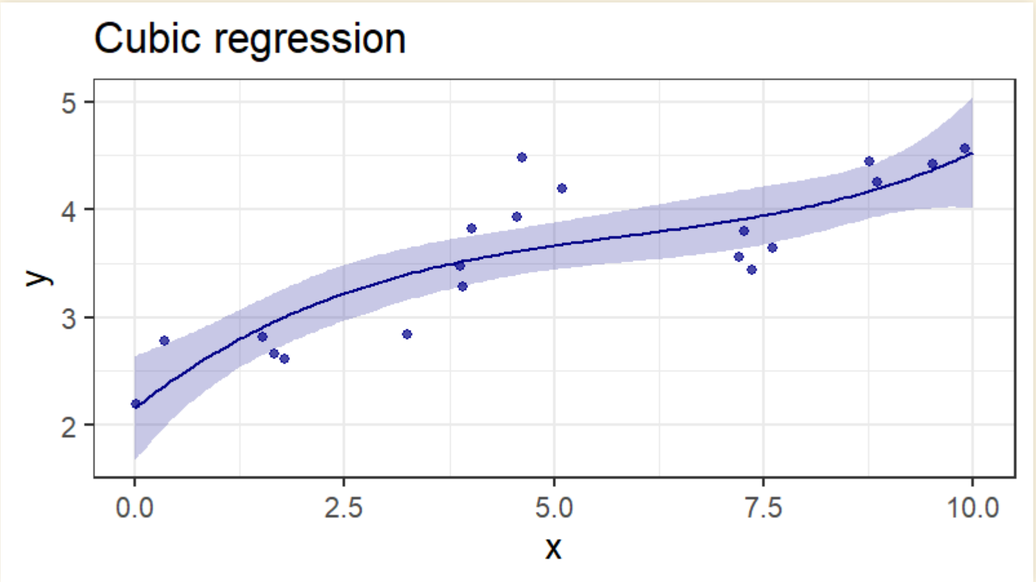
  - Spline

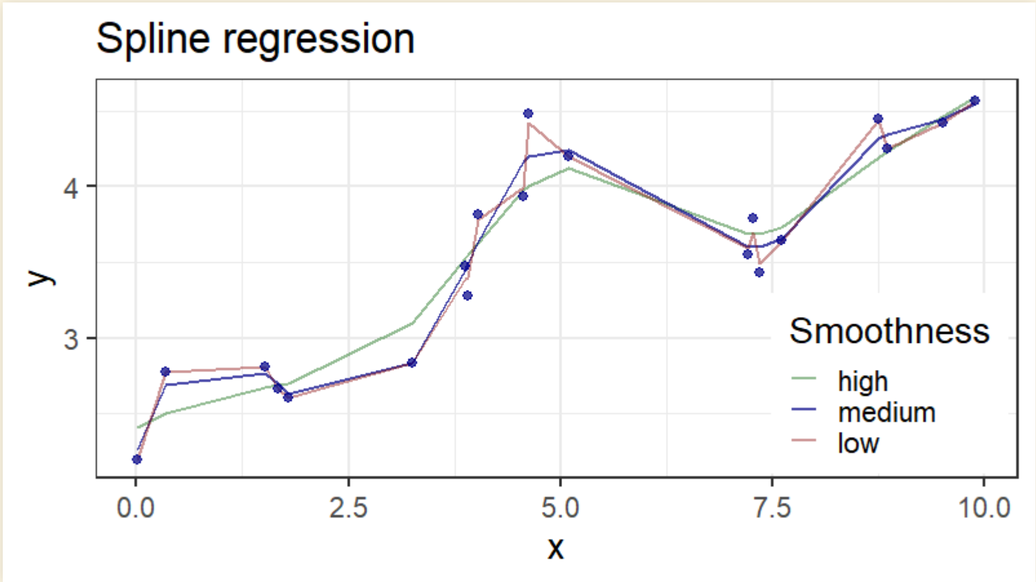


Underfitting vs. Overfitting





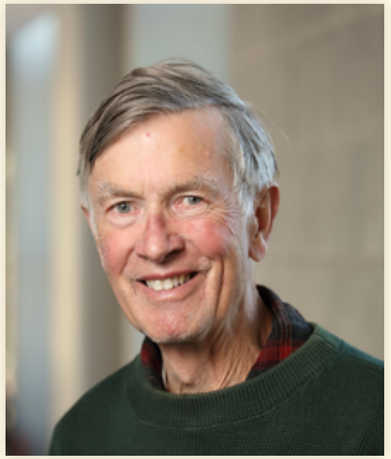




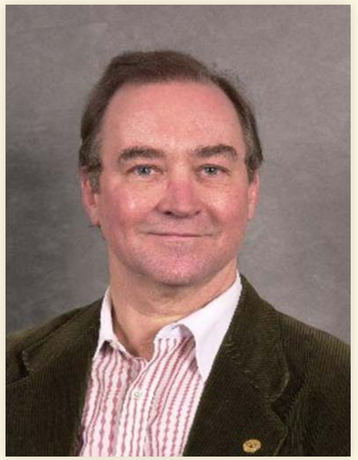
Example from Physics

- Two physicists studying ionization of atoms by microwaves in the late 1980s
  - The spectrum was very complicated
- Tom Gallagher (University of Virginia)
  - Simple model with five parameters.
    - Fit the overall pattern very well, but did not attempt to fit every little bump and wiggle.
- Peter Koch (SUNY Stony Brook)
  - Complicated model:
    - Chaos theory, very hard to understand.
    - Huge computational demands
    - Fit all the details of the spectrum almost perfectly
- Koch and Gallagher spent years fighting fiercely about whose approach to modeling was better.

- Tom Gallagher



- Peter Koch



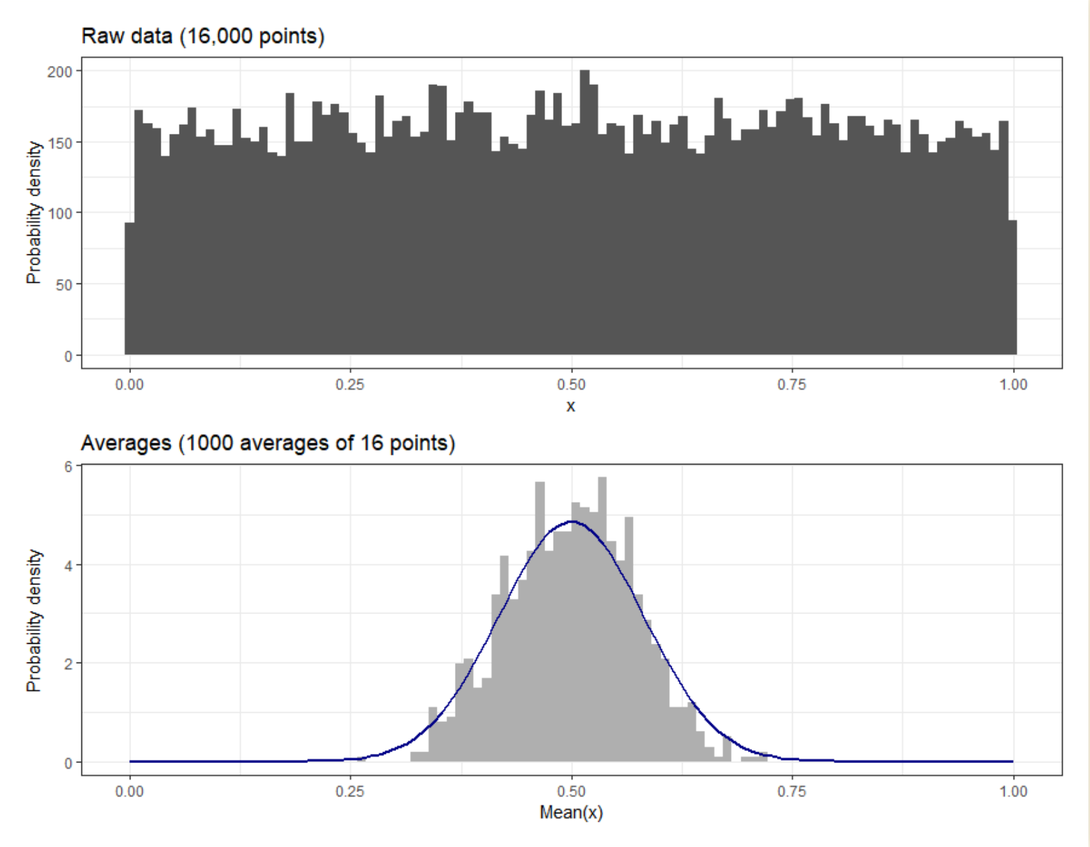
Central Limit Theorem

- Take *N* independent measurements of a variable *x*
  - Some distribution with mean μ, variance σ^2
- In the limit of large *N*, The average will be normal with mean μ and variance σ^2 / N
- This also works for:
  - Product of $(1 + x)$ for small *x*
  - $\log(\text{product}(1 + x))$ for larger *x*

- 1000 replicates of 16 samples each:

```
x <- replicate(1000, runif(16,
          0, 1))
```

mean: 0.50, var: 0.082



A Language for Models

A Recipe for Models

1. Recognize and categorize variables
   - **Data:** Observable variables (x, y)
   - **Parameters:** Latent (unobservable) variables (μ, α, β, σ)
2. Define each variable
   - Functional relations between variables (μ)
   - Probability distributions (y, α, β, σ)
3. Combine variables & relations in a *joint generative model*
   - Analyze real observations (model fitting)
   - Simulate hypothetical observations (predictions)

- Notation for models (example)

$$
\begin{aligned}
y &\sim \text{Normal}(\mu, \sigma) \\
\mu &= \alpha + \beta x \\
\alpha &\sim \text{Normal}(0, 10) \\
\beta &\sim \text{Normal}(0, 10) \\
\sigma &\sim \text{Exponential}(1)
\end{aligned}
$$

- "$\sim$" means a *stochastic* (random *probabilistic*) relationship
  - $x \sim \text{Normal}(\mu, \sigma)$ means that x is a *stochastic* variable, which is drawn at random from a Normal distribution with mean μ and standard deviation σ.

More Model Terminology

- In Bayesian terms:
  - stochastic relationships that define *parameters* are *priors* (α, β, γ, σ)
  - stochastic relationships that define *observed variables* are *likelihoods* that contribute to calculating *posteriors* via Bayes's theorem.
  - When *parameters* (β_i) are defined by *priors* that have their own *parameters* (γ),
    - the *parameters* in the priors are *hyperparameters* (γ),
    - the priors for *hyperparameters* are *hyperpriors*.

- Notation for models (example)

$$y \sim \text{Normal}(\mu, \sigma)$$
$$\mu = \alpha + \sum_i \beta_i x_i$$
$$\alpha \sim \text{Normal}(0, 10)$$
$$\beta_i \sim \text{Normal}(\gamma, 1)$$
$$\gamma \sim \text{Normal}(0, 10)$$
$$\sigma \sim \text{Exponential}(1)$$

Case Study: Height

Height Data

- Anthropometric data
  - !Kung San people
- R data structures:
  - 1-dim: `vector`, `list`
  - 2-dim: `array`, `matrix`, `data.frame`, `tibble` (a kind of `data.frame`)
  - n-dim: `array`, `matrix`
  - `data.frame`, `tibble`:
    - Like a spreadsheet or database:
      - Each column is a variable
        - height, weight, etc.
      - Each row is a set of related measurements
        - height, weight, etc. for a given person

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

data(Howell1)
d <- tibble(Howell1)
head(d)
```

```
## # A tibble: 6 × 4
##   height weight   age  male
##    <dbl>  <dbl> <dbl> <int>
## 1   152.   47.8    63     1
## 2   140.   36.5    63     0
## 3   137.   31.9    65     0
## 4   157.   53.0    41     1
## 5   145.   41.3    51     0
## 6   164.   63.0    35     1
```

```
precis(d)
```

```
##               mean         sd      5.5%     94.5%
## height 138.2635963 27.6024476 81.108550 165.73500
## weight  35.6106176 14.7191782  9.360721  54.50289
## age     29.3443934 20.7468882  1.000000  66.13500
## male     0.4724265  0.4996986  0.000000   1.00000
##              histogram
## height
## weight
## age
## male
```

Cleaning data

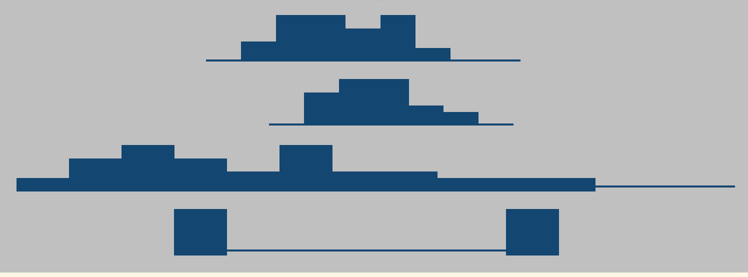
- Many children
  - Children's body proportions are very different to adults'
- Focus on adults
- Model:

$$h \sim \text{Normal}(\mu, \sigma)$$
$$\mu \sim \text{Normal}(178, 20)$$
$$\sigma \sim \text{Uniform}(0, 50)$$

  - Where does Normal(178,20) come from?
- If a quap model gives errors, you may need to give it a hint about where to start looking for the mode of the posterior.

```
d2 <- filter(d, age >= 18)
precis(d2)
```

```
##              mean         sd     5.5%     94.5%
## height 154.59709  7.7423321 142.8750 167.00500
## weight  44.99049  6.4567081  35.1375  55.76588
## age     41.13849 15.9678551  20.0000  70.00000
## male     0.46875  0.4997328   0.0000   1.00000
##              histogram
## height
## weight
## age
## male
```



```
flist <- alist(
    height ~ dnorm(mu, sigma),
    mu ~ dnorm(178, 20),
    sigma ~ dunif(0, 50)
)

mdl_0 <- quap(flist, data = d2)
precis(mdl_0)
```

```
##             mean        sd       5.5%      94.5%
## mu    154.607053 0.4119916 153.948611 155.265495
## sigma   7.731276 0.2913806   7.265593   8.196958
```

```
start <- list(mu = mean(d2$height), sigma = sd(d2$height))
mdl_0a <- quap(flist, data = d2, start = start)
```

Linear Models

- We expect that height is related to other variables (weight, age, sex)
  - Start by looking for an association with weight

$$
\begin{aligned}
h &\sim \text{Normal}(\mu, \sigma) \\
\mu &= \alpha + \beta(w - \bar{w}) \\
\alpha &\sim \text{Normal}(178, 20) \\
\beta &\sim \text{Normal}(0, 10) \\
\sigma &\sim \text{Uniform}(0, 50)
\end{aligned}
$$

where h is height, w is weight, and $\bar{w}$ is the mean weight of all individuals in the sample.

```
wbar <- mean(d2$weight)
flist_1a <- alist(
  height ~ dnorm(mu, sigma),
  mu <- a + b * (weight - wbar),
  a ~ dnorm(178, 20),
  b ~ dnorm(0, 10),
  sigma ~ dunif(0, 50)
)
```

Choosing Priors

Finishing our model

- Our previous model has a prior for β of Normal(0,50).
- Model:
  - β is the relationship between weight and height.
  - Is it really plausible that the slope is as likely to be negative as positive?
  - Plot some examples of the prior
- Try a different prior:
  - Lognormal distribution
    - Good when the parameter must be positive
- Check the variance-covariance matrix
  - Off-diagonal elements are < 0.001:
    - Very little covariance among the parameters. This is good.

$$h \sim \text{Normal}(\mu, \sigma)$$
$$\mu = \alpha + \beta(w - \bar{w})$$
$$\alpha \sim \text{Normal}(178, 20)$$
$$\log(\beta) \sim \text{Normal}(0, 1)$$
$$\sigma \sim \text{Uniform}(0, 50)$$
$$\log(\beta) \sim \text{Normal}(0, 1)$$

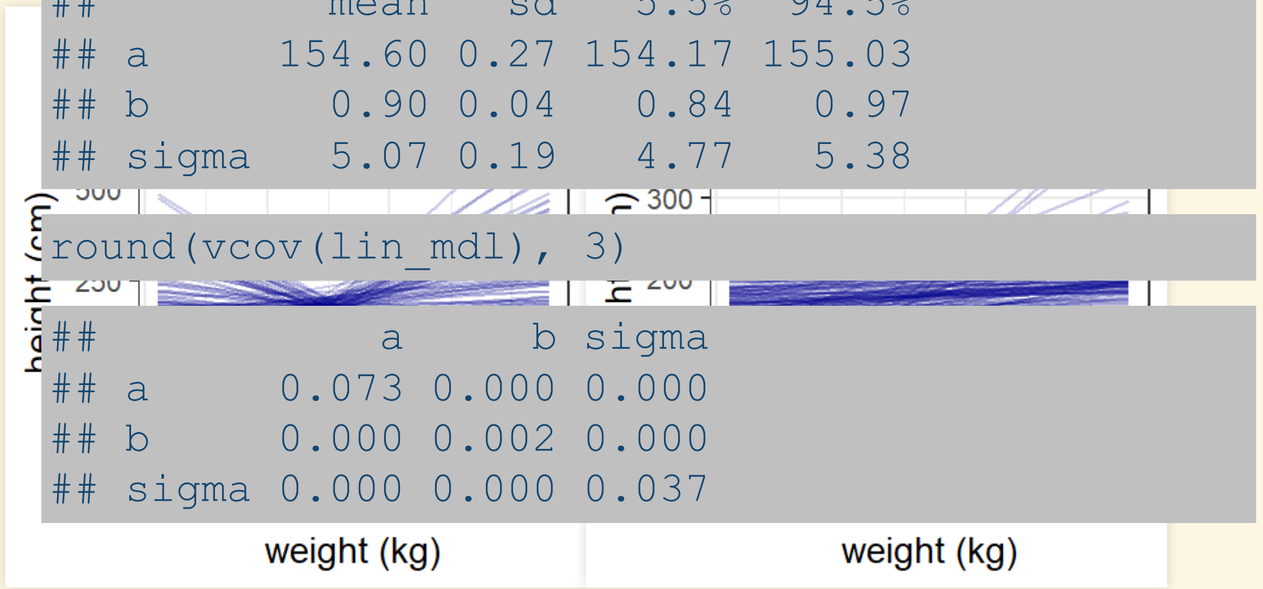
```
b ~ dlnorm(0, 1)
```

```
wbar <- mean(d2$weight)
priors <- tibble(a = rnorm(100, 178, 20), b = rnorm(100, 0,
    10))
flist_lin <- alist(
  height ~ dnorm(mu, sigma),
  mu <- a + b * (weight - wbar),
  a ~ dnorm(178, 20),
  b ~ dlnorm(0, 1),
  sigma ~ dunif(0, 50)
)
lin_mdl <- quap(flist_lin, data = d2)

round(precis(lin_mdl), 2)
```

```
##           mean   sd   5.5%  94.5%
## a       154.60 0.27 154.17 155.03
## b         0.90 0.04   0.84   0.97
## sigma     5.07 0.19   4.77   5.38
```

```
round(vcov(lin_mdl), 3)
```

```
##              a     b sigma
## a        0.073 0.000 0.000
## b        0.000 0.002 0.000
## sigma    0.000 0.000 0.037
```

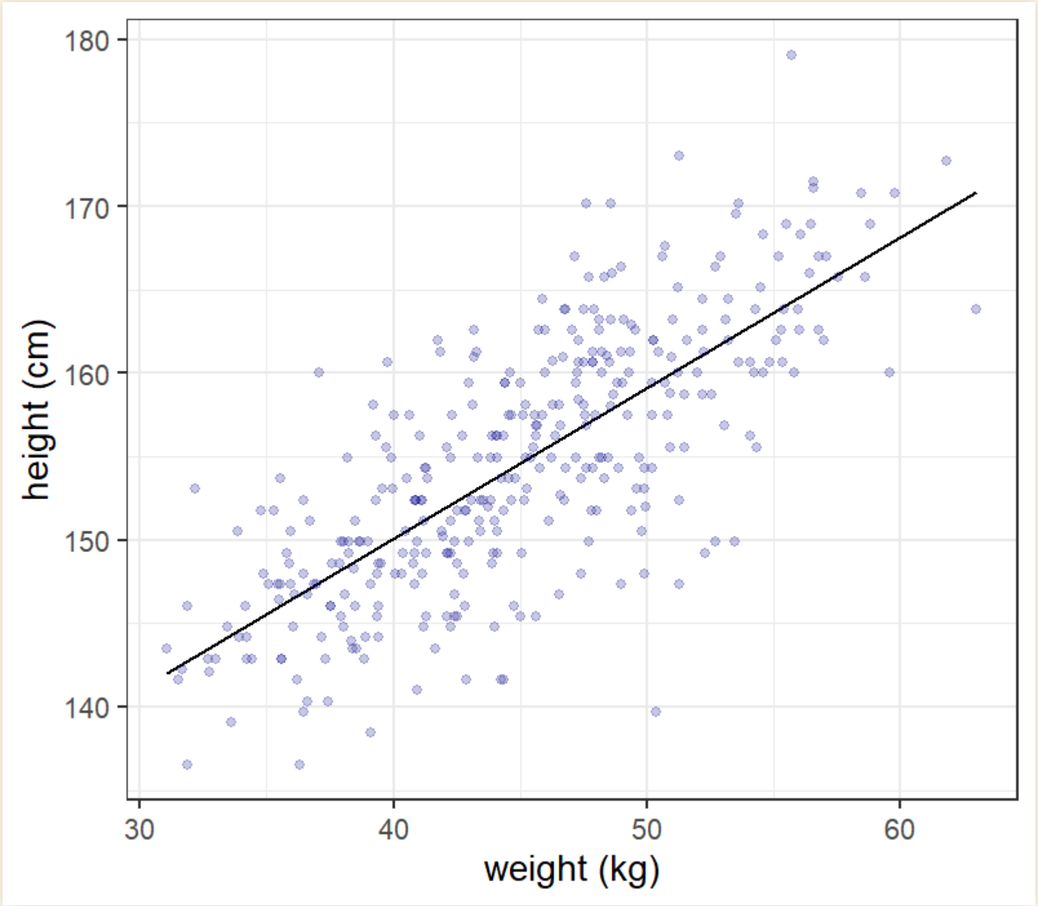


Checking Models

Checking fit

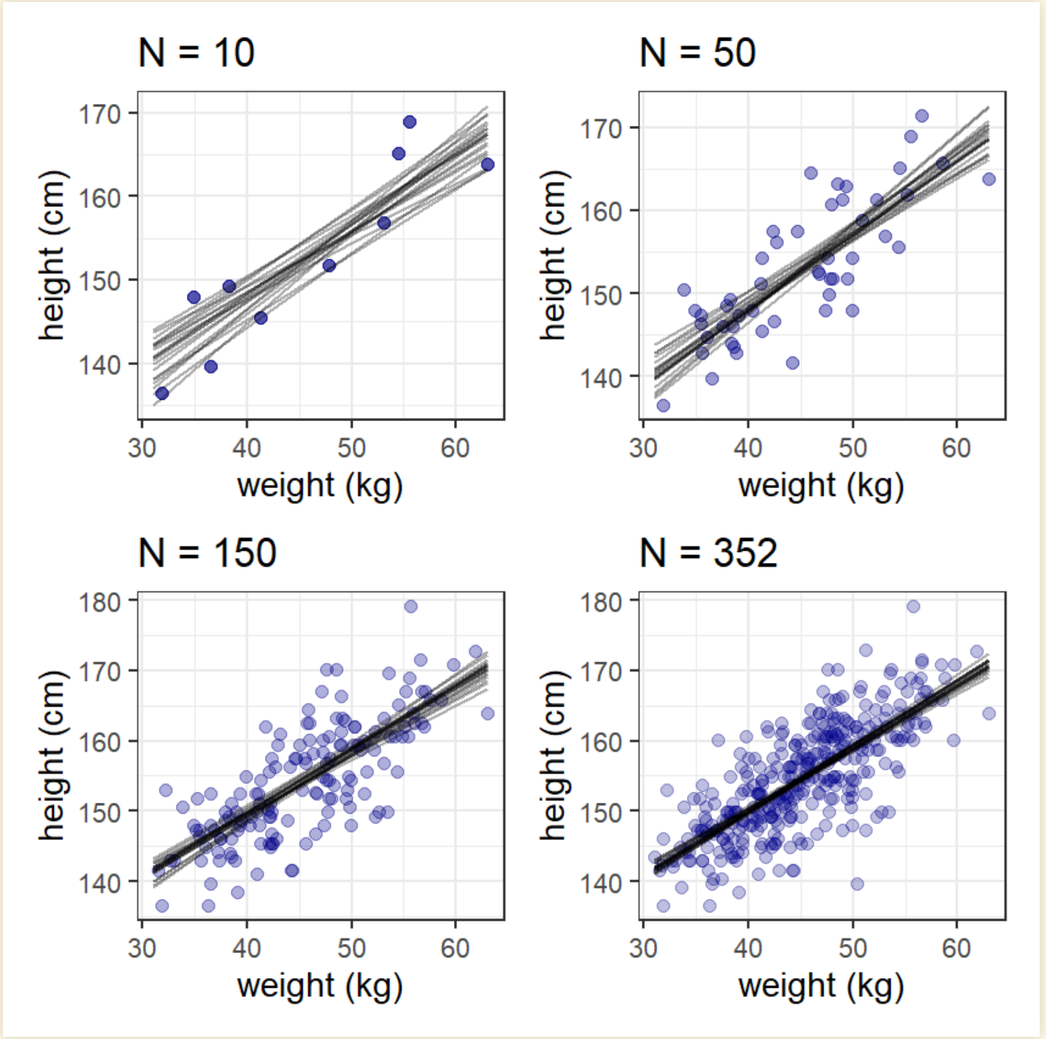
- Sample *a* and *b* from the posterior
- Plot a line with the median slope and intercept
- Plot the original data

```
post <- extract.samples(lin_mdl)
map  <- summarize(post, a = median(a), b = median(b))
map_line <- tibble(weight = range(d2$weight),
                   height = map$a +
                            map$b * (weight - wbar))

ggplot(d2, aes(x = weight, y = height)) +
  geom_line(data = map_line, size = 1, color =
        "black") +
  geom_point(size = 2, color = "darkblue", alpha =
        0.2) +
  labs(x = "weight (kg)", y = "height (cm)")
```



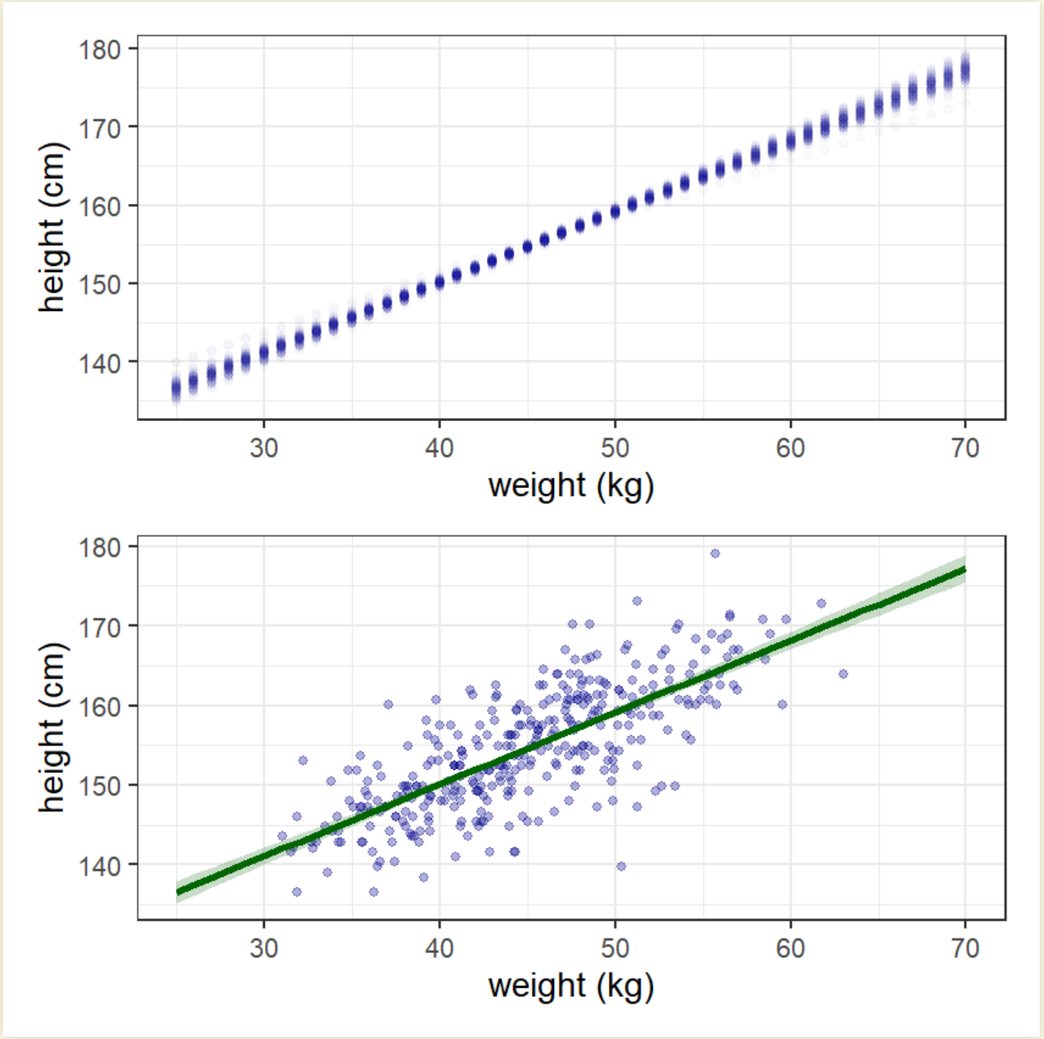
Fit Subsets of Data

```
f <- function(N) {
  dN <- slice_head(d2, n = N)
  mN <- quap(flist_lin, data = dN)
  post <- extract.samples(mN, n = 20)

  map_line <- post |> mutate(index = seq(n())) |>
    cross_join(tibble(weight = range(d2$weight))) |>
    mutate(height = a + b * (weight - wbar))

  ggplot(dN, aes(x = weight, y = height)) +
    geom_point(size = 3, color = "darkblue", alpha = 1.5 / log(N))
        +
    geom_line(data = map_line, mapping = aes(group=index),
              color = "black", alpha = 0.3) +
    labs(x = "weight (kg)", y = "height (cm)",
         title = str_c("N = ", N))
}

library(patchwork)

p1 <- f(10)
p2 <- f(50)
p3 <- f(150)
p4 <- f(nrow(d2))

p1 + p2 + p3 + p4 + plot_layout(ncol = 2)
```



Compatibility Intervals

```
new_data <- tibble(weight = seq(25, 70, 1))
post <- linpred_draws(lin_mdl, new_data, value = "height")
post_sum <- summarize(post, lower = quantile(height, 0.055),
                      upper = quantile(height, 0.945),
                      height = mean(height))

p1 <- ggplot(slice_head(post, n = 100), aes(x = weight, y =
        height)) +
  geom_point(size = 2, alpha = 0.02, color = "darkblue") +
  labs(x = "weight (kg)", y = "height (cm)")

p2 <- ggplot(d2, aes(x = weight, y = height)) +
  geom_point(size = 2, color = "darkblue", alpha = 0.3) +
  geom_smooth(data = post_sum, aes(ymin = lower, ymax = upper),
              stat = "Identity",
              color = "darkgreen", fill = "darkgreen", alpha =
        0.2) +
  labs(x = "weight (kg)", y = "height (cm)")

p1 + p2 + plot_layout(nrow = 2)
```

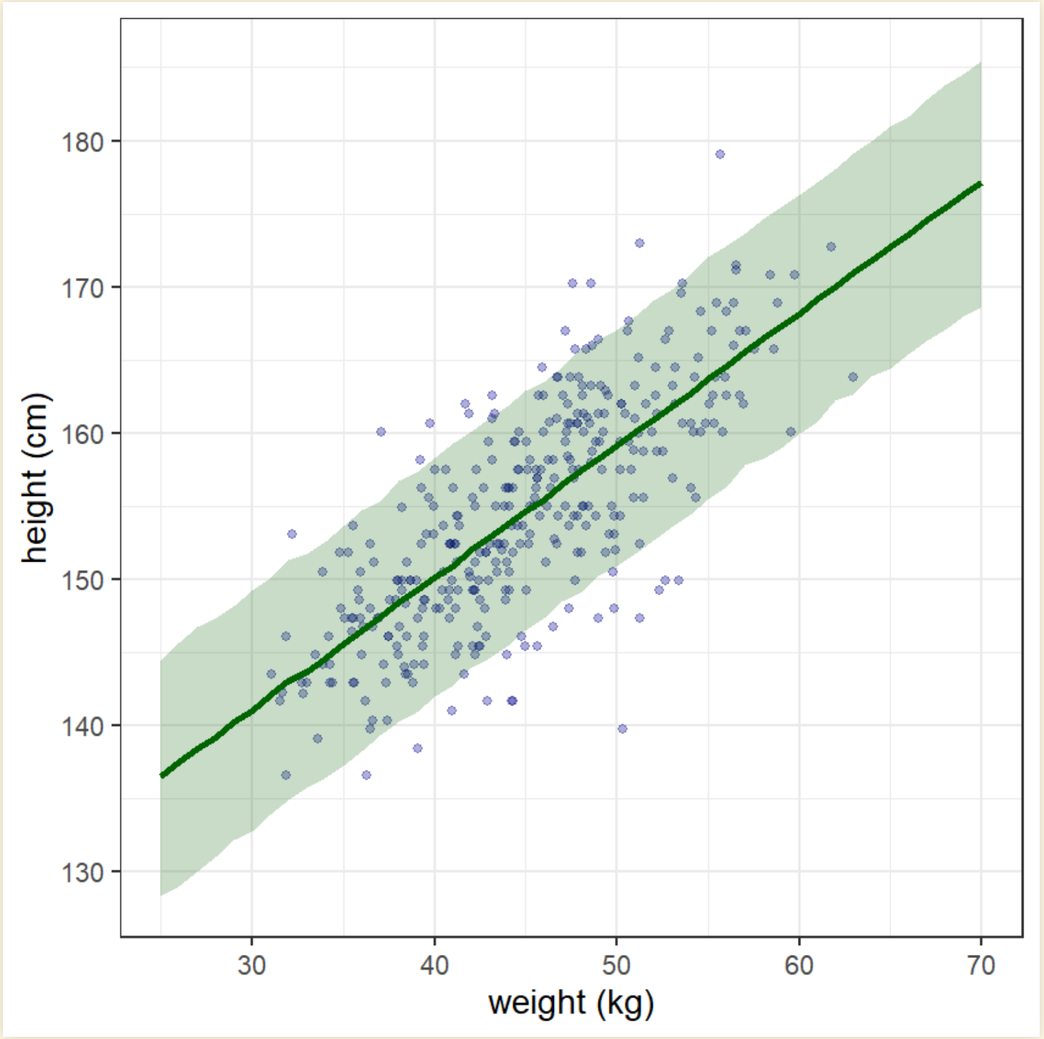


- The top graph shows samples of 100 points drawn from the posterior predictions
- The bottom shows the actual data together with the best-fit line, and the 89% compatibility interval for the line

Checking predictions

```
post <- predicted_draws(lin_mdl, new_data, value = "height")
post_sum <- summarize(post, lower = quantile(height, 0.055),
                      upper = quantile(height, 0.945),
                      height = mean(height))

p2 <- ggplot(d2, aes(x = weight, y = height)) +
  geom_point(size = 2, color = "darkblue", alpha = 0.3) +
  geom_smooth(data = post_sum, aes(ymin = lower, ymax = upper),
              stat = "Identity",
              color = "darkgreen", fill = "darkgreen", alpha =
        0.2) +
  labs(x = "weight (kg)", y = "height (cm)")

p2
```

- The graph shows the actual data together with the best-fit line, and the 89% compatibility interval for the predictions of where we will see data points.
  - 89% of the observed data should lie within this interval.



Beyond Linear Models

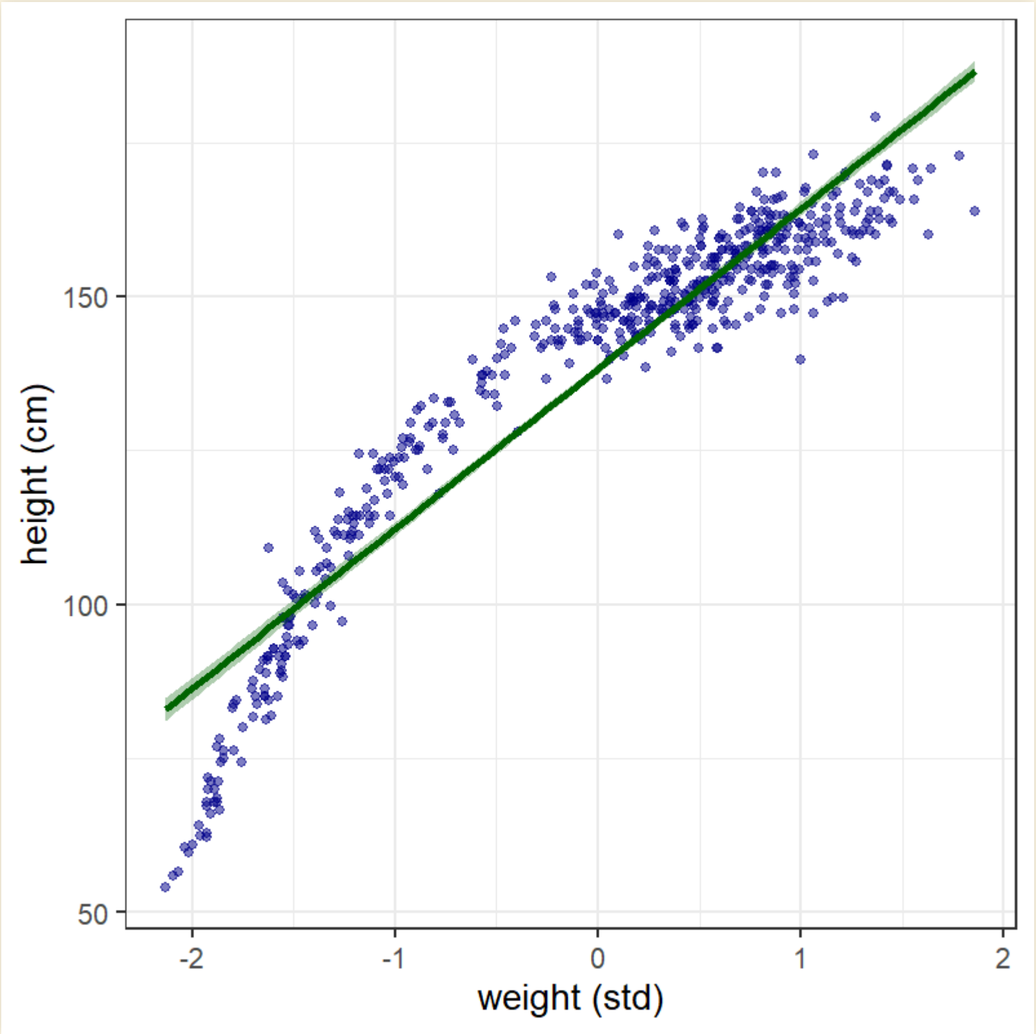
Beyond Linear Models

- Sometimes linear models aren't good enough.
  - If we include children, height is no longer a linear function of weight.
- Polynomial models (quadratic, cubic, etc.) may be better.
- High powers of variables can get large, so **standardize** your data

$$x_s = \frac{x - \bar{x}}{\sigma_x},$$

where $\bar{x}$ is the mean of x and σ_x is the standard deviation.

```
d_std <- mutate(d, ws = (weight - mean(weight)) / sd(weight))

ggplot(d_std, aes(x = ws, y = height)) +
  geom_point(size = 2, color = "darkblue", alpha = 0.5) +
  geom_smooth(method = "lm", color = "darkgreen", fill = "darkgreen",
              alpha = 0.3) +
  labs(x = "weight (std)", y = "height (cm)")
```



Polynomial Regression

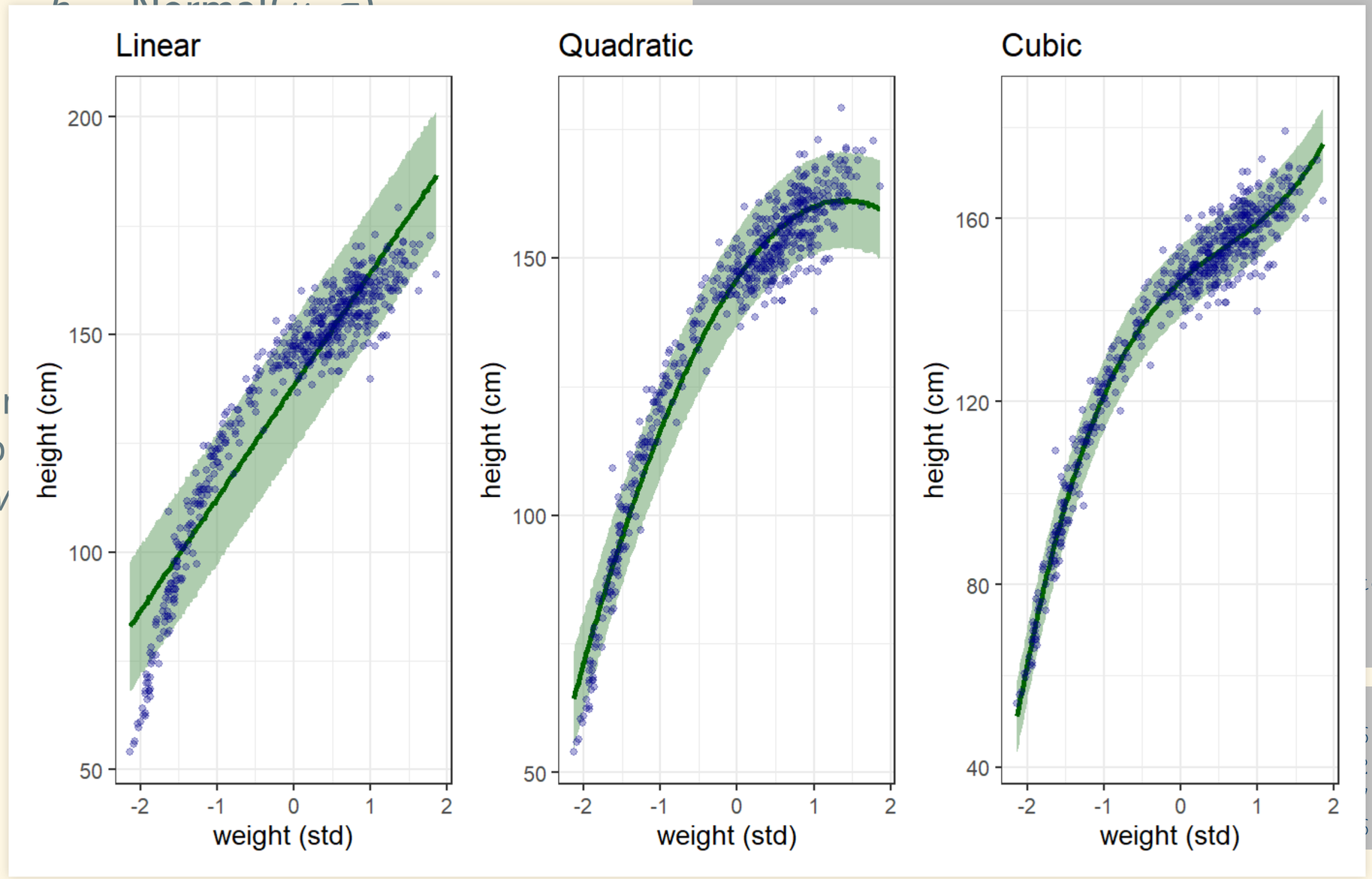
Different polynomials

- Model:

$h \sim \text{Normal}(\mu, \sigma)$

- w_s = stan
- For comp
  variable v

```
d_std <- mutate(d, ws = (weight - mean(weight)) /
                  sd(weight),
                ws2 = ws^2)
```




```
                  94.5%
5.647037
2.194803
7.364975
5.056497
```

Nonparametric Models

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 of cherry tree blossoming dates
    - 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 *i*th point, x_i

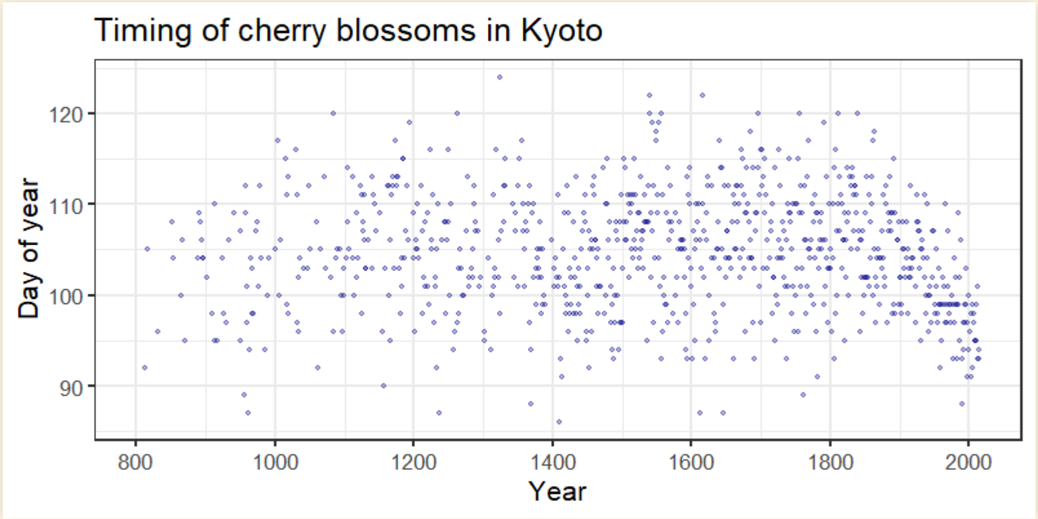
$$\mu_i = \alpha + \sum_{j=1}^{n_{\text{knots}}} w_j B_{i,j}$$

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))
```

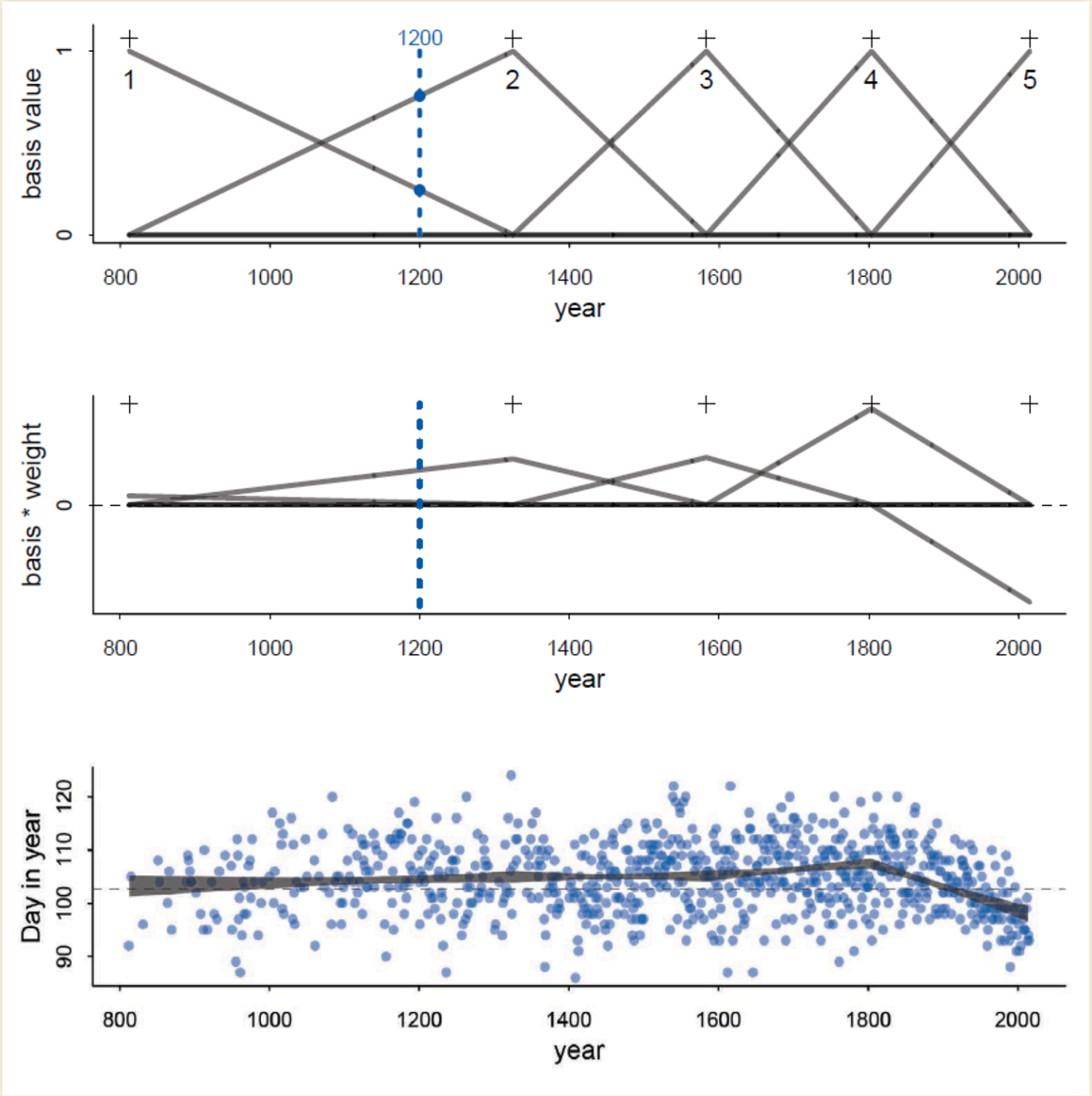
```
## 'data.frame': 1215 obs. of 5 variables:
##                mean     sd   5.5%   94.5%       histogram
## year       1408.00 350.88 867.77 1948.23
## doy         104.54   6.41  94.43  115.00
## temp          6.14   0.66   5.15    7.29
## temp_upper    7.19   0.99   5.90    8.90
## temp_lower   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

- %*% means matrix multiplication