Linear Regression Models
P8111

Lecture 14

Jeff Goldsmith

March 8, 2016

THE DEPARTMENT OF BIOSTATISTICS

Columbia University MAILMAN SCHOOL OF PUBLIC HEALTH

Today's Lecture

- Review session

Exam Thursday

- One hour and twenty minutes (up to two hours)
- Closed book
- Calculators are fine, as long as they don't have internet
- Everything is fair game

What is regression?

"...to understand as far as possible with the available data how the conditional distribution of the response y varies across subpopulations determined by the possible values of the predictor or predictors." – Cook and Weisberg (1999)

Regression model

The process of using data to describe the relationship between outcomes and predictors is called modeling.

- Models are models, not reality.
- "All models are wrong, but some are useful."
- Introduce structure to $f(x)$ to make the problem of estimation easier (this also introduces elements not found in the data, including judgement calls about important features and assumptions about the world).
- We largely focus on *parametric models* $f(x) = f(x; \beta)$ and worry about estimating β.

Linear Regression Models

$f(x;\beta) = \beta_0 + \beta_1 x$

A linear regression model is a particular type of parametric regression.

- Assume $f(x;\beta) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots$
- Focus is on $\beta_0, \beta_1, \dots$
- "Linear" refers to the β's, not the x's:
  - $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2$ is a linear model
  - $f(x) = \beta_0 + x^{\beta_1}$ is not
  - $f^*(x) = \beta_0^* + \beta_1 x^*$

`dplyr` : Two Other Things

- Grouping (`group_by()`) can make some tasks infinitely easier
- The pipe operator (`%>%`) will change your life

Cheat Sheet

Data Wrangling with dplyr and tidyr Cheat Sheet

RStudio

Syntax - Helpful conventions for wrangling

dplyr::tbl_df(iris)

Converts data to tbl class. tbl's are easier to examine than data frames. R displays only the data that fits onscreen:

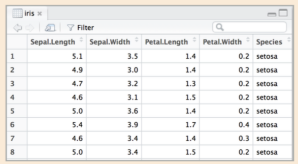
```
Source: local data frame [150 x 5]

   Sepal.Length Sepal.Width Petal.Length
1           5.1         3.5          1.4
2           4.9         3.0          1.4
3           4.7         3.2          1.3
4           4.6         3.1          1.5
5           5.0         3.6          1.4
..          ...         ...          ...
Variables not shown: Petal.Width (dbl),
  Species (fctr)
```

dplyr::glimpse(iris)

Information dense summary of tbl data.

utils::View(iris)

View data set in spreadsheet-like display (note capital V).



dplyr::%>%

Passes object on left hand side as first argument (or . argument) of function on righthand side.

```
x %>% f(y)  is the same as  f(x, y)
y %>% f(x, .., z)  is the same as  f(x, y, z)
```

"Piping" with %>% makes code more readable, e.g.

```
iris %>%
  group_by(Species) %>%
  summarise(avg = mean(Sepal.Width)) %>%
  arrange(avg)
```

Tidy Data - A foundation for wrangling in R

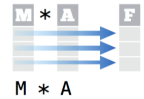
In a tidy data set:

Each **variable** is saved in its own **column**

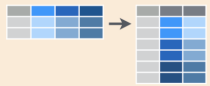
&

Each **observation** is saved in its own **row**

Tidy data complements R's **vectorized operations**. R will automatically preserve observations as you manipulate variables. No other format works as intuitively with R.



Reshaping Data - Change the layout of a data set



tidyr::gather(cases, "year", "n", 2:4)

Gather columns into rows.

tidyr::spread(pollution, size, amount)

Spread rows into columns.

tidyr::separate(storms, date, c("y", "m", "d"))

Separate one column into several.

tidyr::unite(data, col, ..., sep)

Unite several columns into one.

dplyr::data_frame(a = 1:3, b = 4:6)

Combine vectors into data frame (optimized).

dplyr::arrange(mtcars, mpg)

Order rows by values of a column (low to high).

dplyr::arrange(mtcars, desc(mpg))

Order rows by values of a column (high to low).

dplyr::rename(tb, y = year)

Rename the columns of a data frame.

Subset Observations (Rows)

dplyr::filter(iris, Sepal.Length > 7)

Extract rows that meet logical criteria.

dplyr::distinct(iris)

Remove duplicate rows.

dplyr::sample_frac(iris, 0.5, replace = TRUE)

Randomly select fraction of rows.

dplyr::sample_n(iris, 10, replace = TRUE)

Randomly select n rows.

dplyr::slice(iris, 10:15)

Select rows by position.

dplyr::top_n(storms, 2, date)

Select and order top n entries (by group if grouped data).

Logic in R - ?Comparison, ?base::Logic

<	Less than	!=	Not equal to
>	Greater than	%in%	Group membership
==	Equal to	is.na	Is NA
<=	Less than or equal to	!is.na	Is not NA
>=	Greater than or equal to	&, |, !, xor, any, all	Boolean operators

Subset Variables (Columns)

dplyr::select(iris, Sepal.Width, Petal.Length, Species)

Select columns by name or helper function.

Helper functions for select - ?select

select(iris, contains("."))
Select columns whose name contains a character string.

select(iris, ends_with("Length"))
Select columns whose name ends with a character string.

select(iris, everything())
Select every column.

select(iris, matches(".t."))
Select columns whose name matches a regular expression.

select(iris, num_range("x", 1:5))
Select columns named x1, x2, x3, x4, x5.

select(iris, one_of(c("Species", "Genus")))
Select columns whose names are in a group of names.

select(iris, starts_with("Sepal"))
Select columns whose name starts with a character string.

select(iris, Sepal.Length:Petal.Width)
Select all columns between Sepal.Length and Petal.Width (inclusive).

select(iris, -Species)
Select all columns except Species.

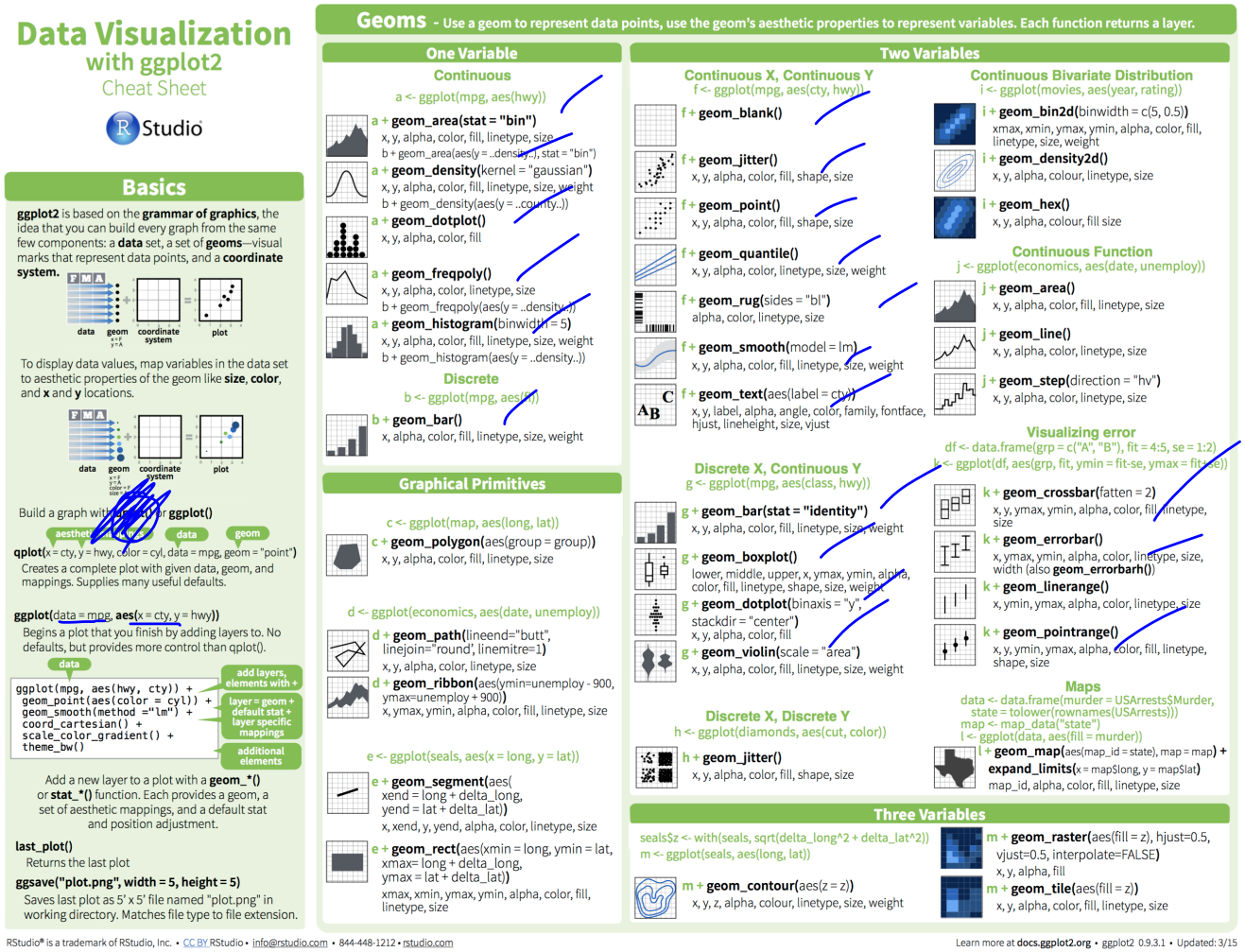
devtools::install_github("rstudio/EDAWR") for data sets

Learn more with browseVignettes(package = c("dplyr", "tidyr")) • dplyr 0.4.0• tidyr 0.2.0 • Updated: 1/15

Constructing a `ggplot` figure

- `data`: the dataframe you're using to construct your plot
- **aesthetic mappings**: connections between data and visual components (x and y, first; size, color, group, shape, etc)
- **layers**: how the data are actually shown (points, lines, boxplots, densities, smooths)

Cheat Sheet


Data Visualization
with ggplot2
Cheat Sheet
RStudio
Basics
ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same few components: a data set, a set of geoms—visual marks that represent data points, and a coordinate system.
data
geom
coordinate system
plot
To display data values, map variables in the data set to aesthetic properties of the geom like size, color, and x and y locations.
Build a graph with qplot() or ggplot()
aesthetic mappings
data
geom
qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.
ggplot(data = mpg, aes(x = cty, y = hwy))
Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().
data
ggplot(mpg, aes(hwy, cty)) +
geom_point(aes(color = cyl)) +
geom_smooth(method ="lm") +
coord_cartesian() +
scale_color_gradient() +
theme_bw()
add layers, elements with +
layer = geom + default stat + layer specific mappings
additional elements
Add a new layer to a plot with a geom_*() or stat_*() function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.
last_plot()
Returns the last plot
ggsave("plot.png", width = 5, height = 5)
Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.
Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.
One Variable
Continuous
a <- ggplot(mpg, aes(hwy))
a + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size
b + geom_area(aes(y = ..density..), stat = "bin")
a + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, linetype, size, weight
b + geom_density(aes(y = ..county..))
a + geom_dotplot()
x, y, alpha, color, fill
a + geom_freqpoly()
x, y, alpha, color, linetype, size
b + geom_freqpoly(aes(y = ..density..))
a + geom_histogram(binwidth = 5)
x, y, alpha, color, fill, linetype, size, weight
b + geom_histogram(aes(y = ..density..))
Discrete
b <- ggplot(mpg, aes(fl))
b + geom_bar()
x, alpha, color, fill, linetype, size, weight
Graphical Primitives
c <- ggplot(map, aes(long, lat))
c + geom_polygon(aes(group = group))
x, y, alpha, color, fill, linetype, size
d <- ggplot(economics, aes(date, unemploy))
d + geom_path(lineend="butt", linejoin="round', linemitre=1)
x, y, alpha, color, linetype, size
d + geom_ribbon(aes(ymin=unemploy - 900, ymax=unemploy + 900))
x, ymax, ymin, alpha, color, fill, linetype, size
e <- ggplot(seals, aes(x = long, y = lat))
e + geom_segment(aes(xend = long + delta_long, yend = lat + delta_lat))
x, xend, y, yend, alpha, color, linetype, size
e + geom_rect(aes(xmin = long, ymin = lat, xmax= long + delta_long, ymax = lat + delta_lat))
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size
Two Variables
Continuous X, Continuous Y
f <- ggplot(mpg, aes(cty, hwy))
f + geom_blank()
f + geom_jitter()
x, y, alpha, color, fill, shape, size
f + geom_point()
x, y, alpha, color, fill, shape, size
f + geom_quantile()
x, y, alpha, color, linetype, size, weight
f + geom_rug(sides = "bl")
alpha, color, linetype, size
f + geom_smooth(model = lm)
x, y, alpha, color, fill, linetype, size, weight
f + geom_text(aes(label = cty))
x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
Discrete X, Continuous Y
g <- ggplot(mpg, aes(class, hwy))
g + geom_bar(stat = "identity")
x, y, alpha, color, fill, linetype, size, weight
g + geom_boxplot()
lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight
g + geom_dotplot(binaxis = "y", stackdir = "center")
x, y, alpha, color, fill
g + geom_violin(scale = "area")
x, y, alpha, color, fill, linetype, size, weight
Discrete X, Discrete Y
h <- ggplot(diamonds, aes(cut, color))
h + geom_jitter()
x, y, alpha, color, fill, shape, size
Continuous Bivariate Distribution
i <- ggplot(movies, aes(year, rating))
i + geom_bin2d(binwidth = c(5, 0.5))
xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight
i + geom_density2d()
x, y, alpha, colour, linetype, size
i + geom_hex()
x, y, alpha, colour, fill size
Continuous Function
j <- ggplot(economics, aes(date, unemploy))
j + geom_area()
x, y, alpha, color, fill, linetype, size
j + geom_line()
x, y, alpha, color, linetype, size
j + geom_step(direction = "hv")
x, y, alpha, color, linetype, size
Visualizing error
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
k <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
k + geom_crossbar(fatten = 2)
x, y, ymax, ymin, alpha, color, fill, linetype, size
k + geom_errorbar()
x, ymax, ymin, alpha, color, linetype, size, width (also geom_errorbarh())
k + geom_linerange()
x, ymin, ymax, alpha, color, linetype, size
k + geom_pointrange()
x, y, ymin, ymax, alpha, color, fill, linetype, shape, size
Maps
data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
l <- ggplot(data, aes(fill = murder))
l + geom_map(aes(map_id = state), map = map) + expand_limits(x = map$long, y = map$lat)
map_id, alpha, color, fill, linetype, size
Three Variables
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
m <- ggplot(seals, aes(long, lat))
m + geom_contour(aes(z = z))
x, y, z, alpha, colour, linetype, size, weight
m + geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE)
x, y, alpha, fill
m + geom_tile(aes(fill = z))
x, y, alpha, color, fill, linetype, size
RStudio® is a trademark of RStudio, Inc. • CC BY RStudio • info@rstudio.com • 844-448-1212 • rstudio.com
Learn more at docs.ggplot2.org • ggplot2 0.9.3.1 • Updated: 3/15

Coefficient interpretation

$E(y|x) = \beta_0 + \beta_1 x$

$E(y|x=0) = \beta_0$ ✓

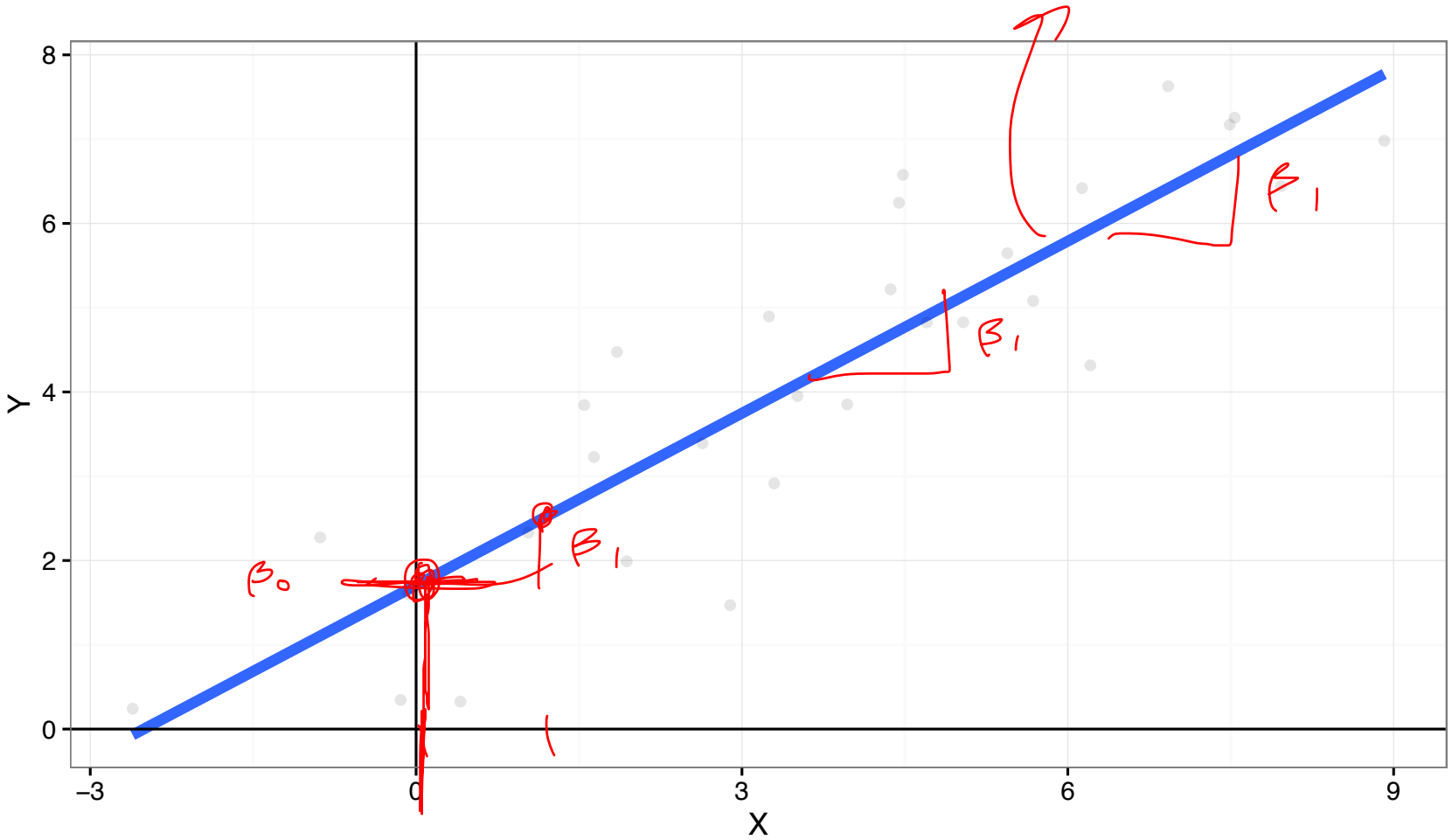
$\beta_1 = (\beta_0 + \beta_1 17) - \beta_0 - \beta_1 16$

$= E(y|x=1) - E(y|x=0)$

17 16

Coefficient interpretation

$$E(y|x) = \beta_0 + \beta_1 x$$



Least squares estimation

$$E(y|x) = \beta_0 + \beta_1 x$$

- Find $\hat{\beta}_0$.

$$RSS(\beta_0, \beta_1) = \sum (y_i - \beta_0 - \beta_1 x_i)^2$$

$$\frac{\partial RSS(\beta_0)}{\partial \beta_0} = -2 \sum (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum y_i - n\beta_0 - \beta_1 \sum x_i = 0$$

$$n\beta_0 = \sum y_i - \beta_1 \sum x_i$$

$$\hat{\beta}_0 = \bar{y} - \beta_1 \bar{x}$$

Least squares estimation

- Now find $\hat{\beta}_1$.

$$RSS(\beta_1) = \sum (y_i - \overbrace{\bar{y} + \beta_1 \bar{x}}^{\hat{\beta}_0} - \beta_1 x_i)^2$$

$$= \sum ((y_i - \bar{y}) - \beta_1 (x_i - \bar{x}))^2$$

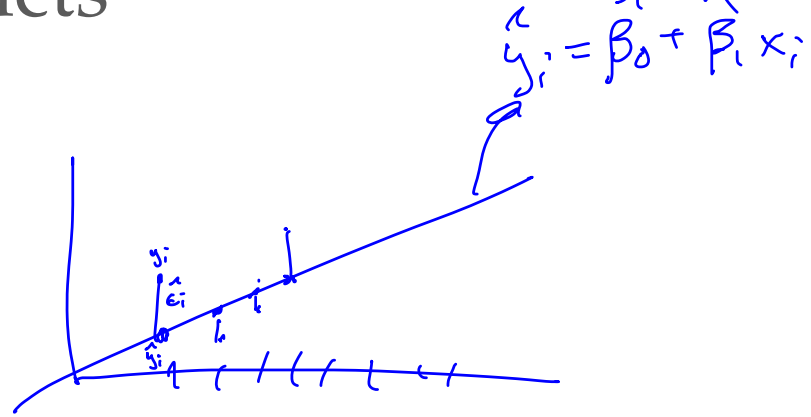
$$\frac{\partial RSS(\beta_1)}{\partial \beta_1} = -2 \sum ((y_i - \bar{y}) - \beta_1 (x_i - \bar{x}))(x_i - \bar{x}) = 0$$

$$= \sum (y_i - \bar{y})(x_i - \bar{x}) - \beta_1 \sum (x_i - \bar{x})^2 = 0$$

$$\hat{\beta}_1 = \frac{\sum (y_i - \bar{y})(x_i - \bar{x})}{\sum (x_i - \bar{x})^2}$$

Some definitions / SLR products

LSE $\hat{\beta}_0, \hat{\beta}_1$

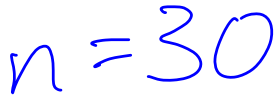


- *Fitted values*: $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$
- *Residuals / estimated errors*: $\hat{\epsilon}_i = y_i - \hat{y}_i$
- *Residual sum of squares*: $\sum_{i=1}^{n} \hat{\epsilon}_i^2$
- *Residual variance*: $\hat{\sigma}^2 = \frac{RSS}{n-2}$
- *Degrees of freedom*: $n - 2$

Notes: residual sample mean is zero; residuals are uncorrelated with fitted values.

R example

```
> summary(linmod)

Call:
lm(formula = y ~ x, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5202 -0.5050 -0.2297  0.5753  1.8534

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.08743    0.22958   9.092 7.53e-10 ***
x            0.61396    0.05415  11.338 5.61e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.8084 on 28 degrees of freedom
Multiple R-squared:  0.8211,    Adjusted R-squared:  0.8148
F-statistic: 128.6 on 1 and 28 DF,  p-value: 5.612e-12
```



Switching to multiple linear regression

- Observe data $(y_i, x_{i1}, \ldots, x_{ip})$ for subjects $1, \ldots, n$. Want to estimate $\beta_0, \beta_1, \ldots, \beta_p$ in the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_1 x_{ip} + \epsilon_i; \quad \epsilon_i \overset{iid}{\sim} (0, \sigma^2)$$

- Assumptions (residuals have mean zero, constant variance, are independent) are as in SLR
- Notation is cumbersome. To fix this, let
  - $\boldsymbol{x}_i = [1, x_{i1}, \ldots, x_{ip}]$
  - $\boldsymbol{\beta}^T = [\beta_0, \beta_1, \ldots, \beta_p]$
  - Then $y_i = \boldsymbol{x}_i \boldsymbol{\beta} + \epsilon_i$

Matrix notation

- Let

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad X = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_p \end{bmatrix}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

- Then we can write the model in a more compact form:

$$\underset{n\times 1}{\boldsymbol{y}} = \underset{n\times(p+1)}{\boldsymbol{X}}\underset{(p+1)\times 1}{\boldsymbol{\beta}} + \underset{n\times 1}{\boldsymbol{\epsilon}}$$

- X is called the *design matrix*

Matrix notation

$$E\left(\begin{bmatrix}\epsilon_1\\ \vdots\\ \epsilon_n\end{bmatrix}\right) = \begin{bmatrix}0\\ \vdots\\ 0\end{bmatrix} \qquad \epsilon_i \overset{iid}{\sim} (0, \sigma^2) \qquad Var\left(\begin{bmatrix}\epsilon_1\\ \vdots\\ \epsilon_n\end{bmatrix}\right) = \begin{bmatrix}\sigma^2 & \cdots & 0\\ 0 & \sigma^2 & \\ \vdots & & \ddots\\ 0 & & \sigma^2\end{bmatrix}$$

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

- $\boldsymbol{\epsilon}$ is a random vector rather than a random variable
- $E(\boldsymbol{\epsilon}) = 0$ and $Var(\boldsymbol{\epsilon}) = \sigma^2 I$
- Note that *Var* is potentially confusing; in the present context it means the "variance-covariance matrix"

Interpretation of coefficients

$\beta_1 =$ $(\beta_0 + \beta_1) - (\beta_0)$

$= (\beta_0 + \beta_1 + 17\beta_2 + 4.3\beta_3 \ldots) - (\beta_0 + \beta_2 17 + \beta_3 4.3 \ldots)$

$E(y \mid x_1 = 1, x_2 = 17, \ldots) - E(y \mid x_1 = 0, x_2 = 17)$

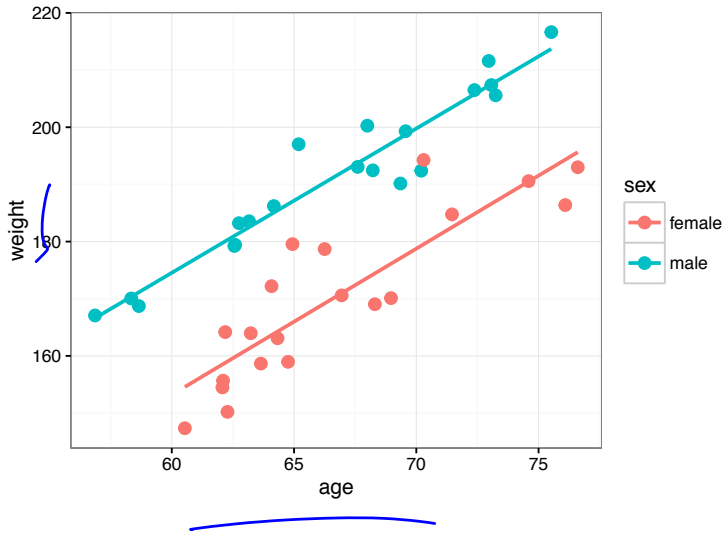
β_1 = the diff in $E(y)$ for a 1-unit Δx_1

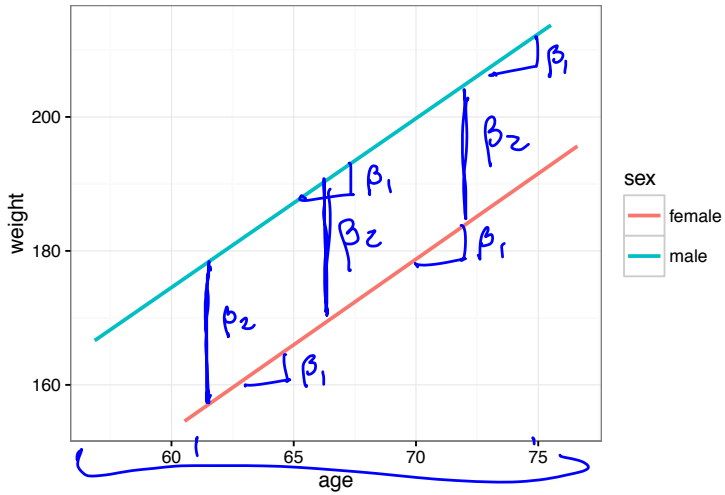
Keeping everything else fixed!

Example with two predictors

$$E(y|x) = \beta_0 + \beta_1 x_{age} + \beta_2 x_{sex}$$

$$\beta_0 = E(y \mid age = 0,\ sex = male)$$





Interpretation of coefficients

$x_{i,sex} = 0$

$$y_i = \beta_0 + \beta_1 x_{i,age} + \epsilon_i$$

$\beta_0 = E(y \mid age = 0, men)$

$\beta_1 = \Delta E(y)$ for a 1 unit Δ age for men

$x_{i,sex} = 1$

$$y_i = \underbrace{(\beta_0 + \beta_2)}_{*} + \underbrace{(\beta_1 + \beta_3)}_{*} x_{i,age} + \epsilon_i$$

* $\beta_0 + \beta_1 x_{i,age} + \beta_2 + \beta_3 x_{i,age}$

$\beta_0 + \beta_2$ = Fit for women

$\beta_1 + \beta_3$ = Age slope for women

$$\beta_2 = (\beta_0 + \beta_2) - (\beta_0)$$

$$\beta_3 = \underbrace{(\beta_1 + \beta_3)}_{\uparrow} - (\beta_1)$$

Indicator variables

- Let x be a categorical variable with k levels (e.g. with $k = 3$ "low", "med", "high").
- Choose one group as the baseline (e.g. "low")
- Create $(k-1)$ binary terms to include in the model:

$$\begin{aligned} x_{\text{med},i} &= I(x_i = \text{“med”}) \\ x_{\text{high},i} &= I(x_i = \text{“high”}) \end{aligned}$$

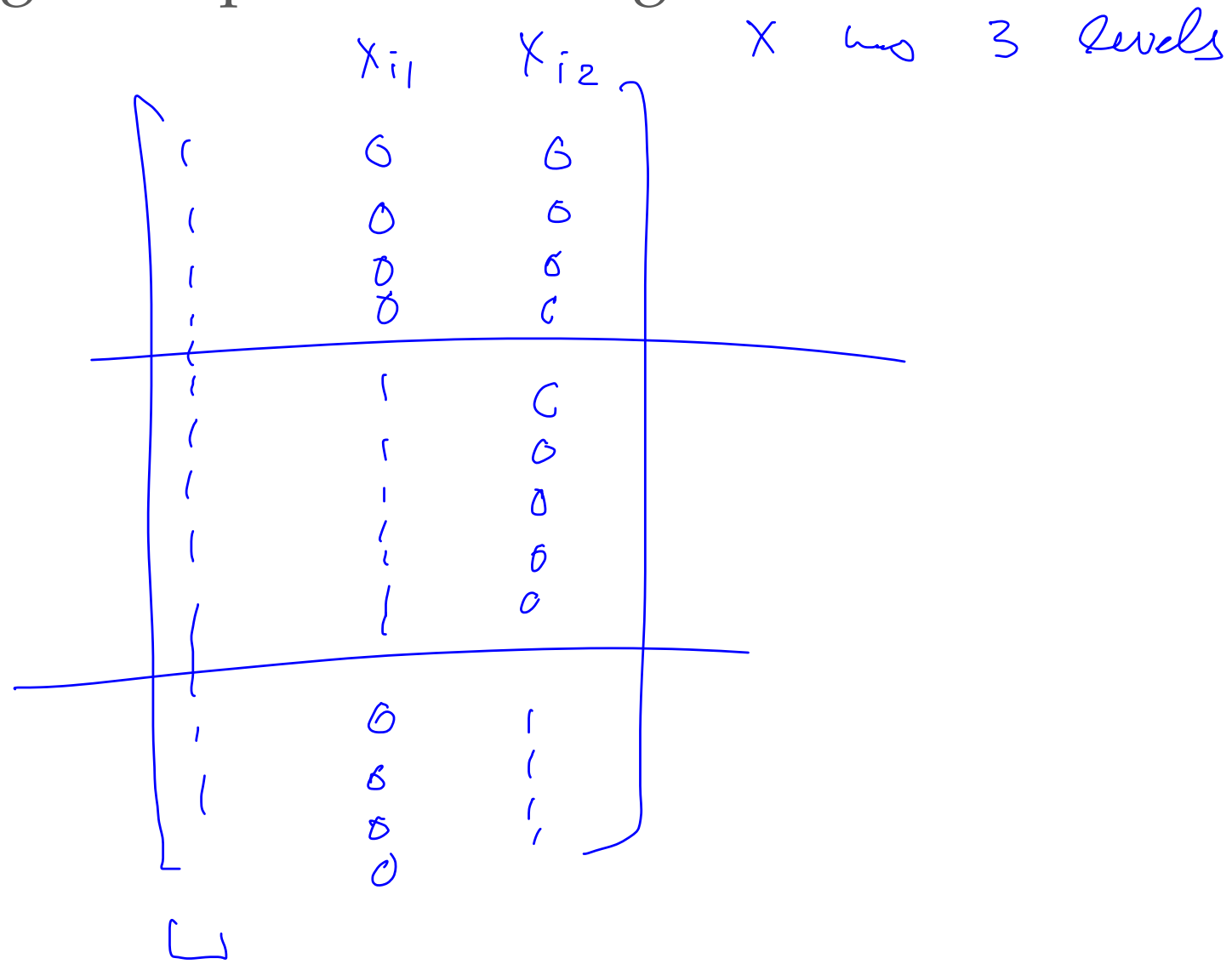
$x_{med} = 0$, $x_{high} = 0$ $\Rightarrow$ low!

- For a model with no additional predictors, pose the model

$$y_i = \beta_0 + \beta_1 x_{i1} + \ldots + \beta_{k-1} x_{i,k-1} + \epsilon_i$$

  and estimate parameters using least squares
- Note distinction between *predictors* and *terms*

Categorical predictor design matrix



Polynomial models

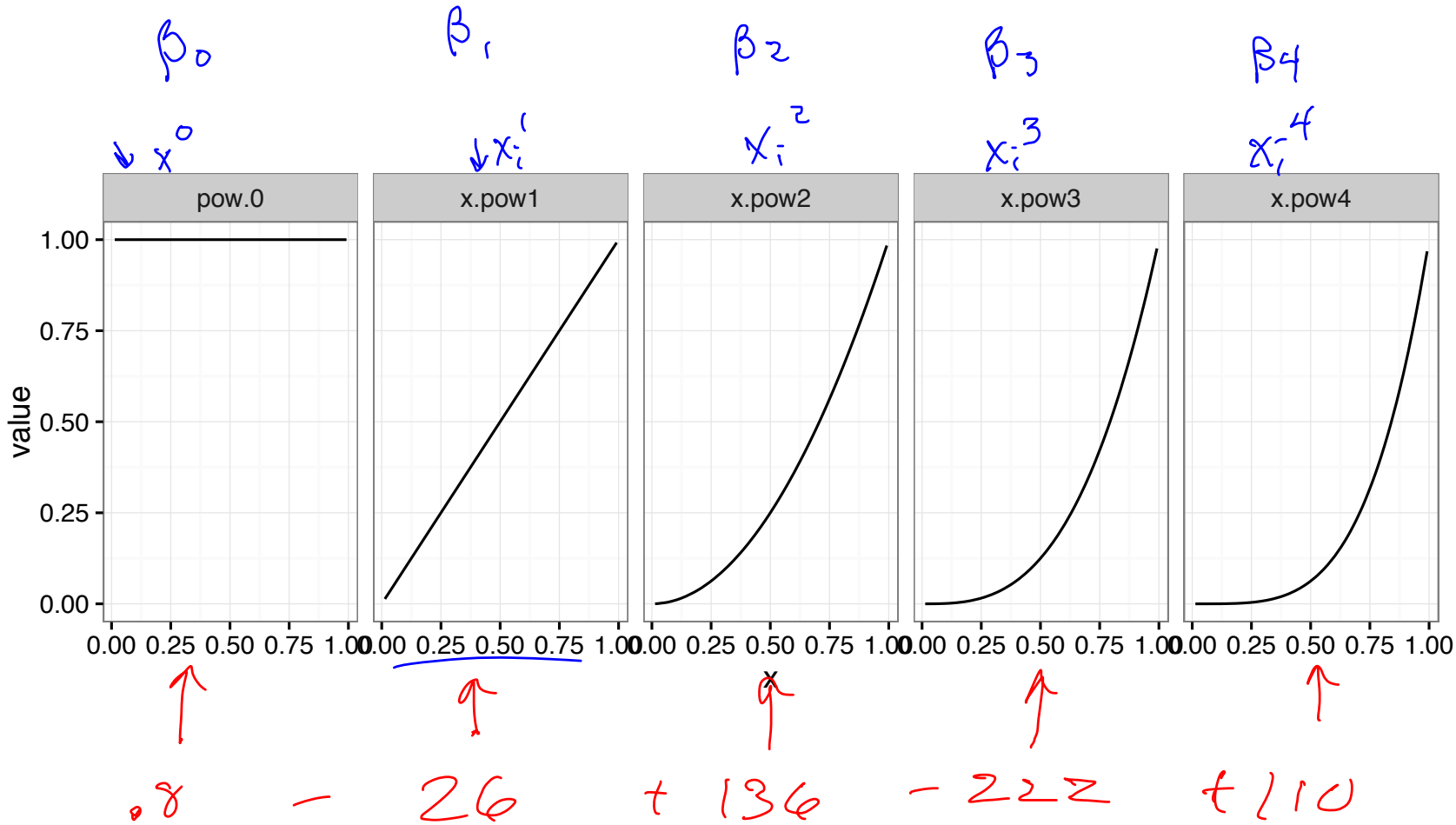
$$y_i = x_i \beta + \epsilon_i$$

- Model of the form

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots + \beta_p x_i^p + \epsilon_i;\ \epsilon_i \overset{iid}{\sim} (0, \sigma^2)$$

- p is the polynomial order
- More polynomial terms can lead to a better approximation of $E(y|x)$, but also higher variability in the fit
- Conversely, smaller p can lead to inability to capture $E(y|x)$, but is often more stable
- Quadratic fits are pretty okay. I don't trust cubic and beyond.

Polynomial models



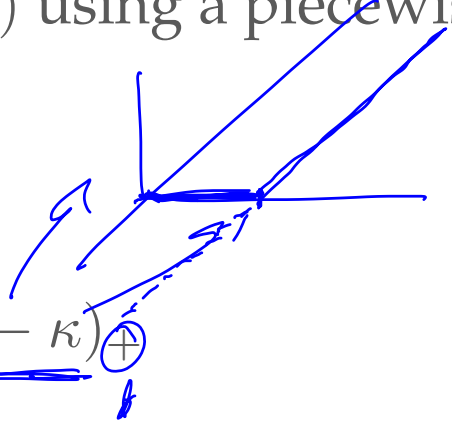
Piecewise linear models

Suppose we want to estimate $\mathrm{E}(y|x) = f(x)$ using a piecewise linear model.

- For one knot we can write this as

$$\mathrm{E}(y|x) = \beta_0 + \beta_1 x + \beta_2 (x - \kappa)_+$$

where κ is the location of the change point



Interpretation of regression coefficients

$y_i = \beta_0 + \beta_1 x_i + \beta_2 (x_i - k)_+ + \epsilon_i$

$x < k$

$E(y|x+1) - E(y|x)$

$= \beta_0 + \beta_1(x+1) - \beta_0 - \beta_1 x$

$= \beta_0 - \beta_0 + \beta_1 x - \beta_1 x + \beta_1$

$\beta_1 = E\Delta y$ for a 1 unit Δx, $x < k$

$x > k$

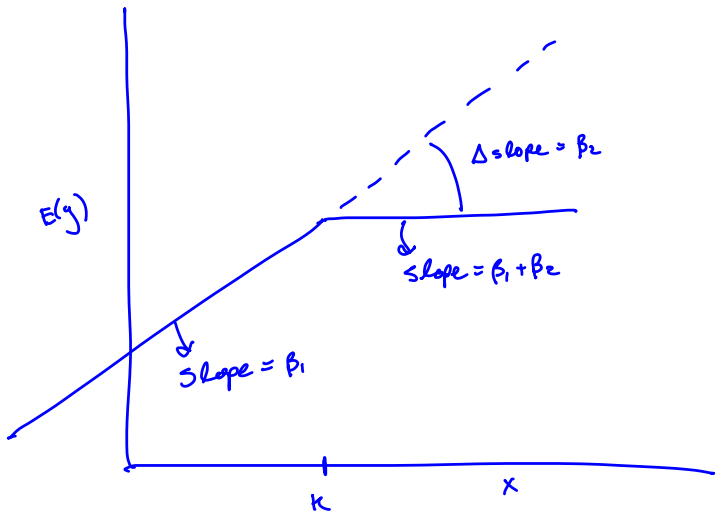
$E(y|x+1) - E(y|x)$

$= \beta_0 + \beta_1(x+1) + \beta_2(x+1-k) - \beta_0 - \beta_1 x - \beta_2(x-k)$

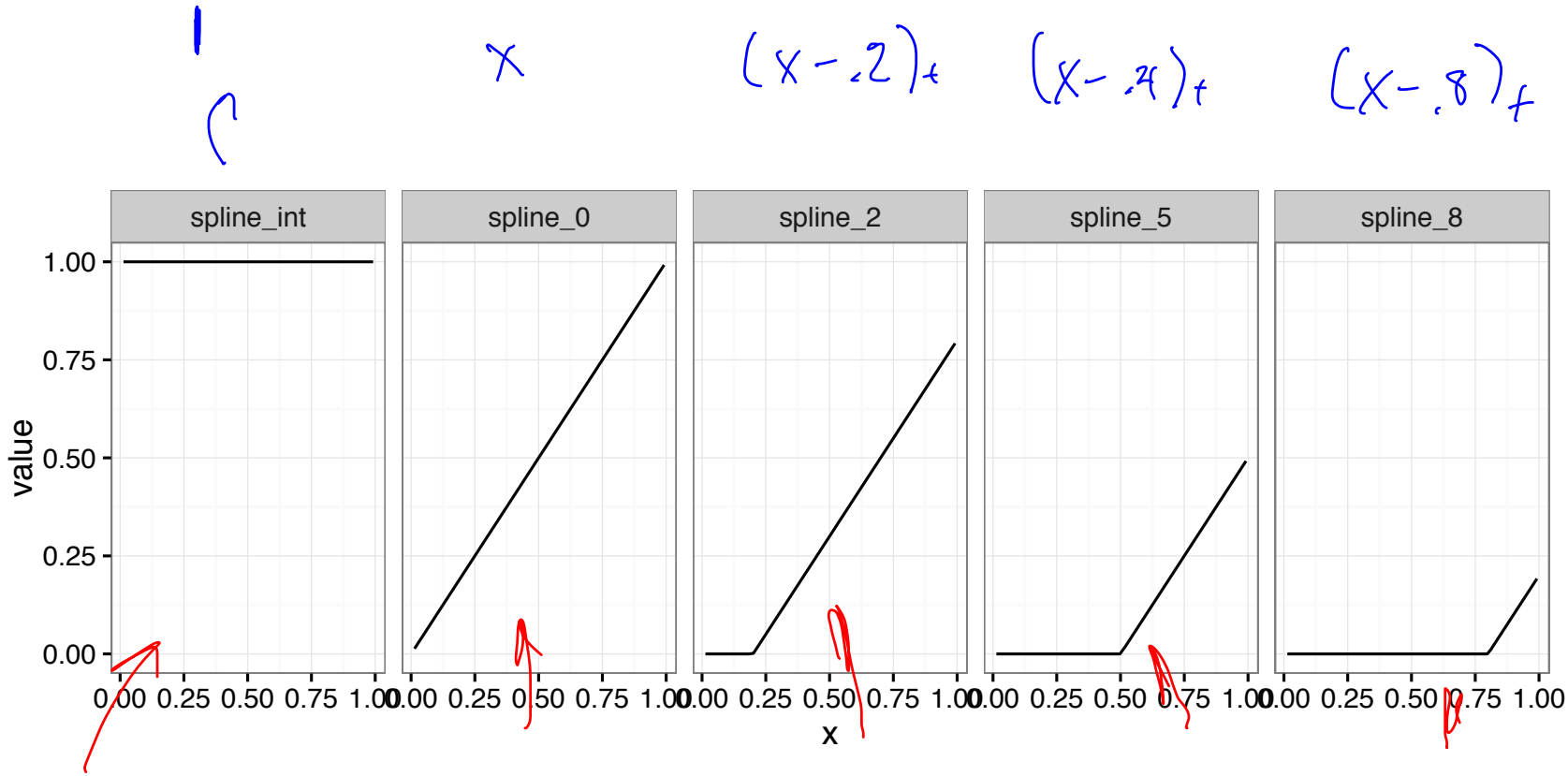
$= \beta_0 - \beta_0 + \beta_1 x - \beta_1 x + \beta_1 + \beta_2 x - \beta_2 x + \beta_2 k - \beta_2 k + \beta_2$

$\beta_1 + \beta_2 = E\Delta y$ for a 1 unit Δx, $x > k$

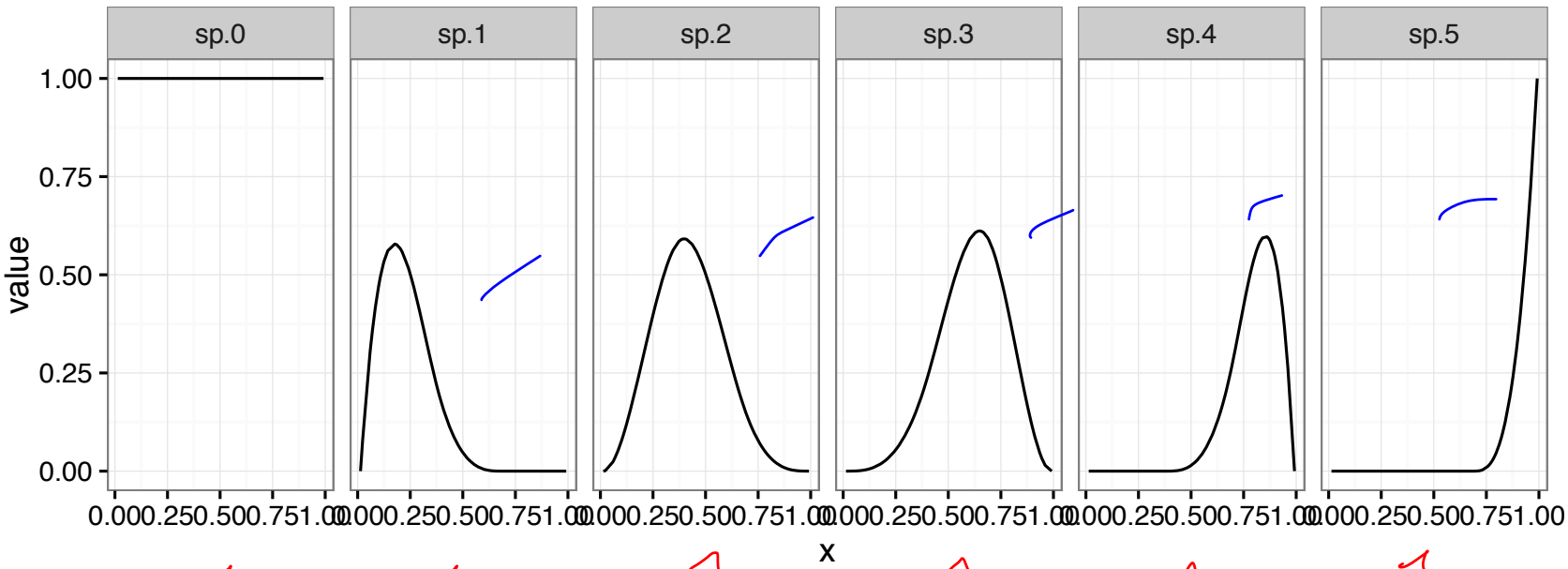
β_2 = change in slope, comparing $x > k$ to $x < k$



Example


spline_int
spline_0
spline_2
spline_5
spline_8
value
x
1.00
0.75
0.50
0.25
0.00
0.00 0.25 0.50 0.75 1.00

Spline models



Least squares

As in simple linear regression, we want to find the β that minimizes the residual sum of squares.

$RSS(\beta) = \sum_i \epsilon_i^2 =$ $\begin{bmatrix} \epsilon_1 & \epsilon_2 & \epsilon_3 & \dots & \epsilon_n \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix}$

$\epsilon^T \epsilon$

$(y - X\beta)^T (y - X\beta)$

Least squares

$$\frac{\partial RSS(\beta)}{\partial \beta} = \begin{cases} (y - X\beta)^T (y - X\beta) = \\ (y^T - \beta^T X^T)(y - X\beta) \\ \beta^T X^T X \beta - \beta^T X^T y - y^T X \beta + y^T y \\ (\beta^T X^T X \beta - 2\beta^T X^T y + y^T y) \\ RSS(\beta) \end{cases}$$

$$(X^T X + (X^T X)^T)\beta - 2X^T y$$

$$2X^T X\beta - 2X^T y = 0$$

$$\boxed{(X^T X)\beta = X^T y}$$

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

Unbiasedness of LSEs

$$E(\hat{\boldsymbol{\beta}}) =$$

$y = x\beta + \epsilon$

$\epsilon \sim (0, \sigma^2 I)$

$y \sim (x\beta, \sigma^2 I)$

$E\left((x^T x)^{-1} x^T y\right)$

$= (x^T x)^{-1} x^T E(y)$

$= (x^T x)^{-1} x^T x \beta$

$= \beta$

Variance of LSEs

$$Var(\hat{\boldsymbol{\beta}}) =$$

$$Var((X^TX)^{-1}X^Ty)$$

$$= (X^TX)^{-1}X^T \, Var(y) \, X(X^TX)^{-1}$$

$$= \sigma^2 (X^TX)^{-1}(X^TX)(X^TX)^{-1}$$

$$= \sigma^2 (X^TX)^{-1}$$

$$\sigma^2 \begin{bmatrix} \square & \square \\ \downarrow & \square \\ Cov(\beta_0, \beta_q) & \square \end{bmatrix}$$

$$\widehat{Var}(\hat{\beta}) = \hat{\sigma}^2 (X^TX)^{-1}$$

$$Var(c\hat{\boldsymbol{\beta}}) =$$

$$c \, Var(\hat{\beta}) \, c^T$$

$$\sigma^2 c (X^TX)^{-1} c^T$$

$$\begin{bmatrix} 1 & x_{i1} & x_{i2} & x_{i3} \end{bmatrix} \begin{bmatrix} \\ \uparrow \\ c^T \\ \\ \end{bmatrix} = \hat{y}_i$$

Definitions

"Hat matrix"

- *Fitted values*: $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{H}\boldsymbol{y}$
- *Residuals / estimated errors*: $\hat{\boldsymbol{\epsilon}} = \boldsymbol{y} - \hat{\boldsymbol{y}}$
- *Residual sum of squares*: $\sum_{i=1}^{n} \hat{\epsilon}_i^2 = \hat{\boldsymbol{\epsilon}}^T\hat{\boldsymbol{\epsilon}}$
- *Residual variance*: $\hat{\sigma}^2 = \frac{RSS}{n-p-1}$
- *Degrees of freedom*: $n - p - 1$

$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$

pred other than intercept

Hat matrix

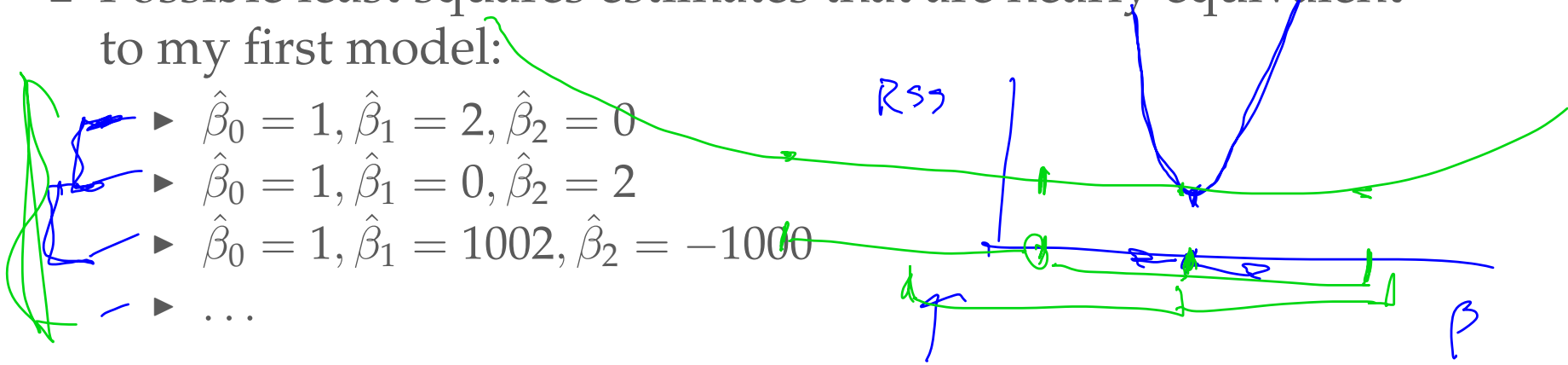
Some properties of the hat matrix:

- It is a projection matrix: $\boldsymbol{HH} = \boldsymbol{H}$
- It is symmetric: $\boldsymbol{H}^T = \boldsymbol{H}$
- The residuals are $\hat{\epsilon} = (\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}$ $y - Hy = y - \hat{y}$
- The inner product of $(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{y}$ and $\boldsymbol{Hy}$ is zero (predicted values and residuals are uncorrelated).

Effects of collinearity

Suppose I fit a model $y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i$.

- I have estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 2$
- I put in a new variable $x_2 = x_1 + \textit{error}$, where *error* is pretty small
- My new model is $y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$
- Possible least squares estimates that are nearly equivalent to my first model:
  - $\hat{\beta}_0 = 1, \hat{\beta}_1 = 2, \hat{\beta}_2 = 0$
  - $\hat{\beta}_0 = 1, \hat{\beta}_1 = 0, \hat{\beta}_2 = 2$
  - $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1002, \hat{\beta}_2 = -1000$
  - $\ldots$
- A unique solution exists, but it is hard to find



Sampling distribution of $\hat{\beta}$

$\hat{\beta} \sim (\beta, \sigma^2(x^Tx)^{-1})$

If our usual assumptions are satisfied and $\epsilon \sim \mathrm{N}\left[0, \sigma^2 I\right]$ then

$$\hat{\boldsymbol{\beta}} \sim \mathrm{N}\left[\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^T\boldsymbol{X})^{-1}\right].$$

$y \sim N(x\beta, \sigma^2 I)$

- This will be used later for inference.
- Even without Normal errors, asymptotic Normality of LSEs is possible under reasonable assumptions.

Asymptotic distribution

Assume that

- $E(\epsilon_i|\boldsymbol{x}_i) = 0 \; \forall i$;
- $Var(\epsilon_i|\boldsymbol{x}_i) = \sigma^2 \; \forall i$;
- $n \overset{lim}{\to} \infty \; \frac{\mathbf{X}^T\mathbf{X}}{n} \to Q$ where Q is a finite non-singular matrix.

Then

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \to \mathrm{N}\left[0, \sigma^2 Q^{-1}\right]$$

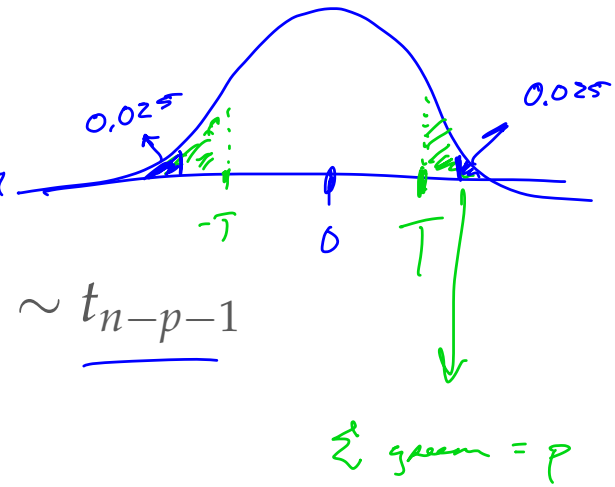
(This is essentially an extension of the central limit theorem)

Individual coefficients

For individual coefficients

- We can use the test statistic

$$T = \frac{\hat{\beta}_j - \beta_j}{\widehat{se}(\hat{\beta}_j)} = \frac{\hat{\beta}_j - \beta_j}{\sqrt{\hat{\sigma}^2 (\mathbf{X}^T \mathbf{X})^{-1}_{jj}}} \sim t_{n-p-1}$$



- For a two-sided test of size α, we reject if

$$|T| > t_{1-\alpha/2, n-p-1}$$

- The p-value gives $P(t_{n-p-1} > T_{obs} | H_0)$

Note that t is a symmetric distribution that converges to a Normal as $n - p - 1$ increases.

Inference for linear combinations

Sometimes we are interested in making claims about $c^T\beta$ for some c.

- Define $H_0 : c^T\beta = c^T\beta_0$ or $H_0 : c^T\beta = 0$
- We can use the test statistic

$$T = \frac{c^T\hat{\beta} - c^T\beta_0}{\widehat{se}(c^T\hat{\beta})} = \frac{c^T\hat{\beta} - c^T\beta_0}{\sqrt{\hat{\sigma}^2 c^T(\mathbf{X}^T\mathbf{X})^{-1}c}}$$

- This test statistic is asymptotically Normally distributed
- For a two-sided test of size α, we reject if

$$|T| > z_{1-\alpha/2}$$

Global tests

$H_0: \beta_1 = \beta_2 = 0$

$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \ldots + \epsilon_i$

$y_i = \beta_0 + 0 + 0 + \beta_3 x_{i3} + \ldots + \epsilon_i$

Compare a smaller "null" model to a larger "alternative" model

- Smaller model must be nested in the larger model
- That is, the smaller model must be a special case of the larger model
- For both models, the RSS gives a general idea about how well the model is fitting
- In particular, something like

$$\frac{RSS_S - RSS_L}{RSS_L}$$

small = Reject

β_{ij} = fail to Reject

compares the relative RSS of the models

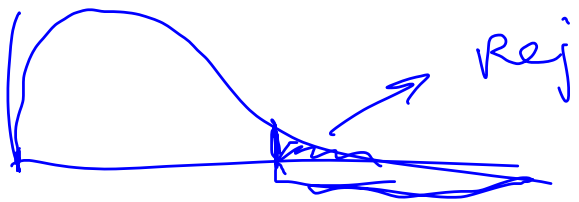
Global F tests

- Compute the test statistic

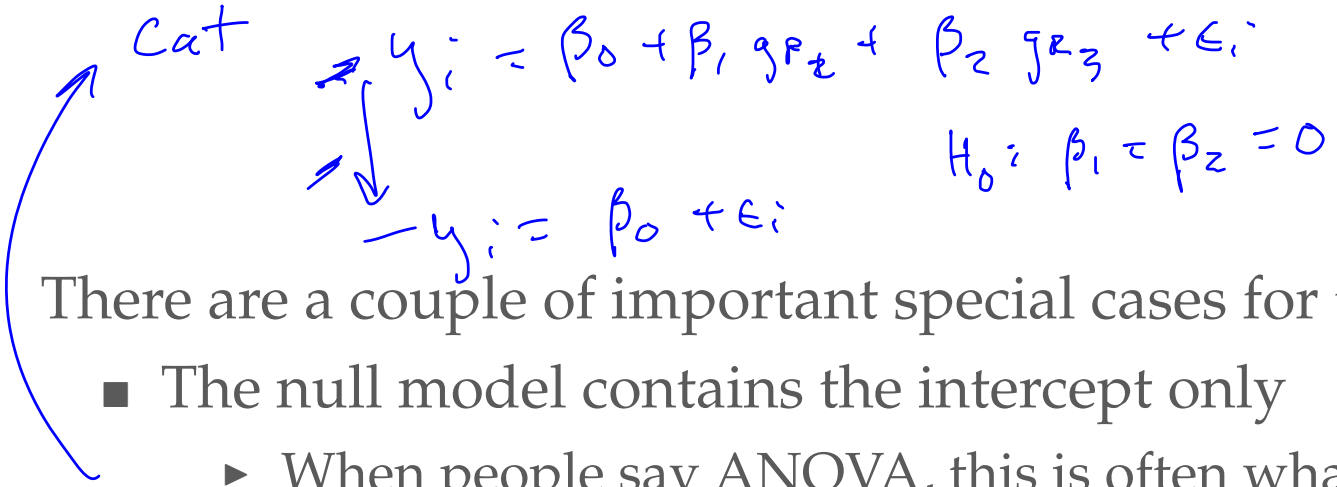
$$F_{obs} = \frac{(RSS_S - RSS_L)/(df_S - df_L)}{RSS_L/df_L}$$

- If H_0 (the null model) is true, then $F_{obs} \sim F_{df_S - df_L, df_L}$
- Note $df_s = n - p_S - 1$ and $df_L = n - p_L - 1$
- We reject the null hypothesis if the p-value is above α, where

$$\text{p-value} = P(F_{df_S - df_L, df_L} > F_{obs})$$



Global F tests



There are a couple of important special cases for the F test

- The null model contains the intercept only
  - When people say ANOVA, this is often what they mean (although all F tests are based on an analysis of variance)
- The null model and the alternative model differ only by one term
  - Gives a way of testing for a single coefficient
  - Turns out to be equivalent to a two-sided t-test: $t^2_{df_L} \sim F_{1,df_L}$

$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$

$y_i = \beta_0 + \beta_1 x_{i1} + \epsilon_i$

$H_0: \beta_2 = 0$

Alternative global tests: the Wald test

For a vector of coefficients, we can test $H_0 : \boldsymbol{\beta} = \boldsymbol{\beta}_0$:

- Use the test statistic

$$W = (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^T [\widehat{Var}(\hat{\boldsymbol{\beta}})]^{-1} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)$$

- Under the null, this test statistic has an asymptotic χ^2_p distribution
- In practice, we replace $Var(\hat{\boldsymbol{\beta}})$ with $\widehat{Var}(\hat{\boldsymbol{\beta}})$ and use an F distribution

Alternative global tests: the likelihood ratio test

$H_0: \beta_2 = \beta_3 = 0$

If we are using maximum likelihood estimation (we'll cover this soon – turns out to be least squares in MLR), we can use a LRT:

- Use the test statistics

$$\Delta = -2 \log \frac{L_0}{L_1} = -2(l_0 - l_1)$$

Likelihood for null / for alt

- This test statistic has an asymptotic χ^2_d distribution where d is the difference in the number of parameters between the two models.
- Must compare nest models

Confidence intervals: individual parameters

- A confidence interval with coverage $(1-\alpha)$ is given by

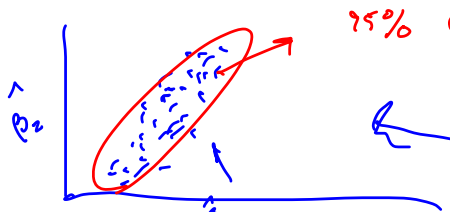
$$\hat{\beta}_j \pm t_{1-\alpha/2,n-p-1}\widehat{se}(\hat{\beta}_j)$$

- Assuming all the standard assumptions hold,

$$(1-\alpha) \text{ "=" } P(LB < \beta_j < UB)$$

Note there is a one-to-one correspondence between this confidence interval and the hypothesis test.

Confidence intervals: multiple parameters



$\forall \beta_0 \in CI$, fail to reject

$\left(\hat{\beta}_j + 2\,\widehat{se}(\hat{\beta}_j) \right)$

↳ inverting t test

We might want a confidence region for multiple coefficients simultaneously

$CR(\beta_1, \beta_2)$

- Invert Wald test for multiple coefficients – find region containing all values β_0 for which p-value from global Wald test is $> \alpha$

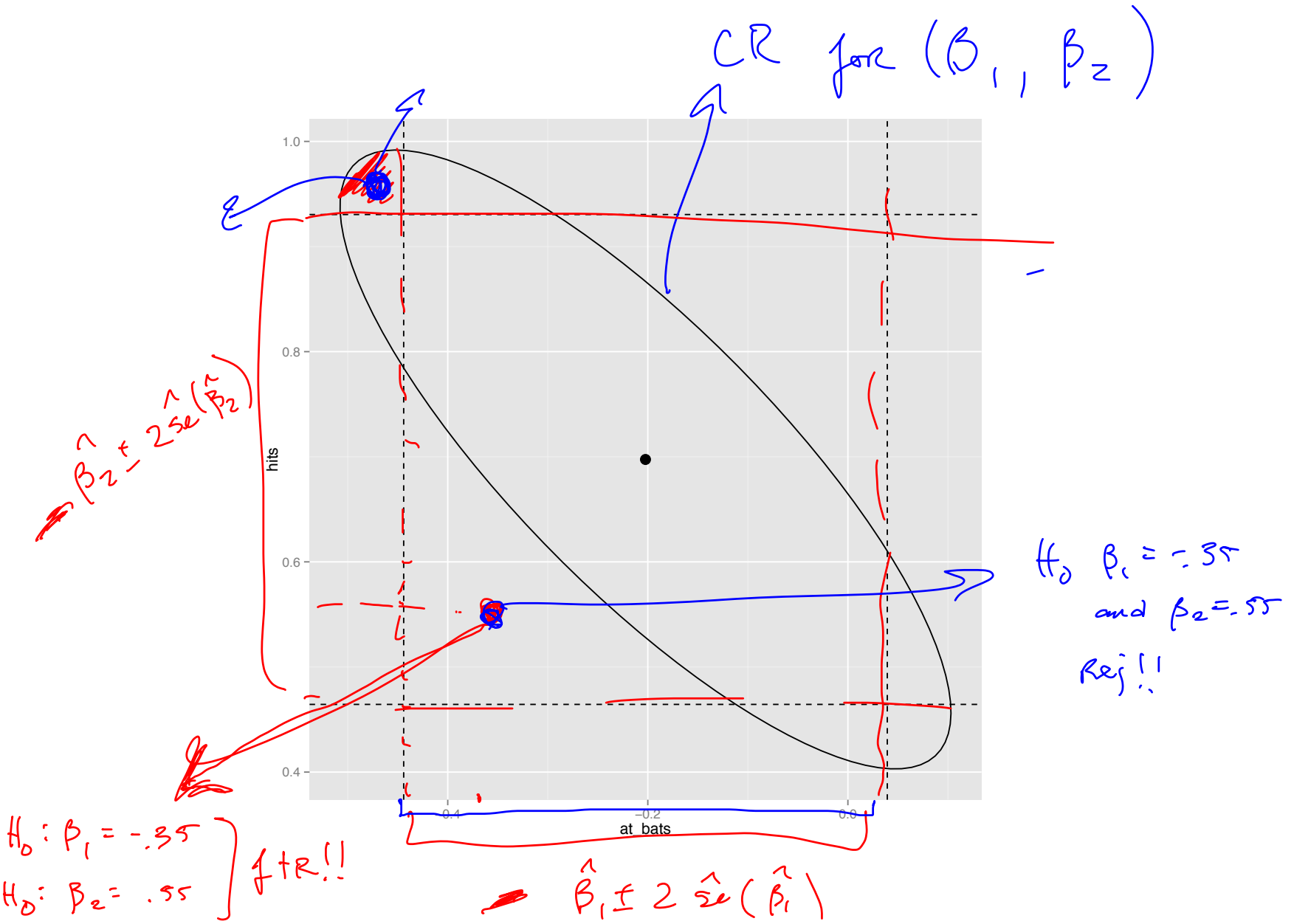
↓ $\forall \beta_0$, $(\hat{\beta} - \beta_0)^T (Var(\hat{\beta})^{-1}) (\hat{\beta} - \beta_0)$ $\leq$ Critical Value

- Then

$$(1-\alpha) \text{ "=" } P[\beta \in \text{region}]$$

- This region is an ellipsoid in higher dimensions; we can visualize in 2D most easily and 3D pretty well.

Confidence intervals: multiple parameters



Predictions and prediction intervals

- What is the prediction value y for a given x_0
- What range would you give for the value of a new outcome?
- Two sources of variance to consider: variance in estimates and variance in outcome

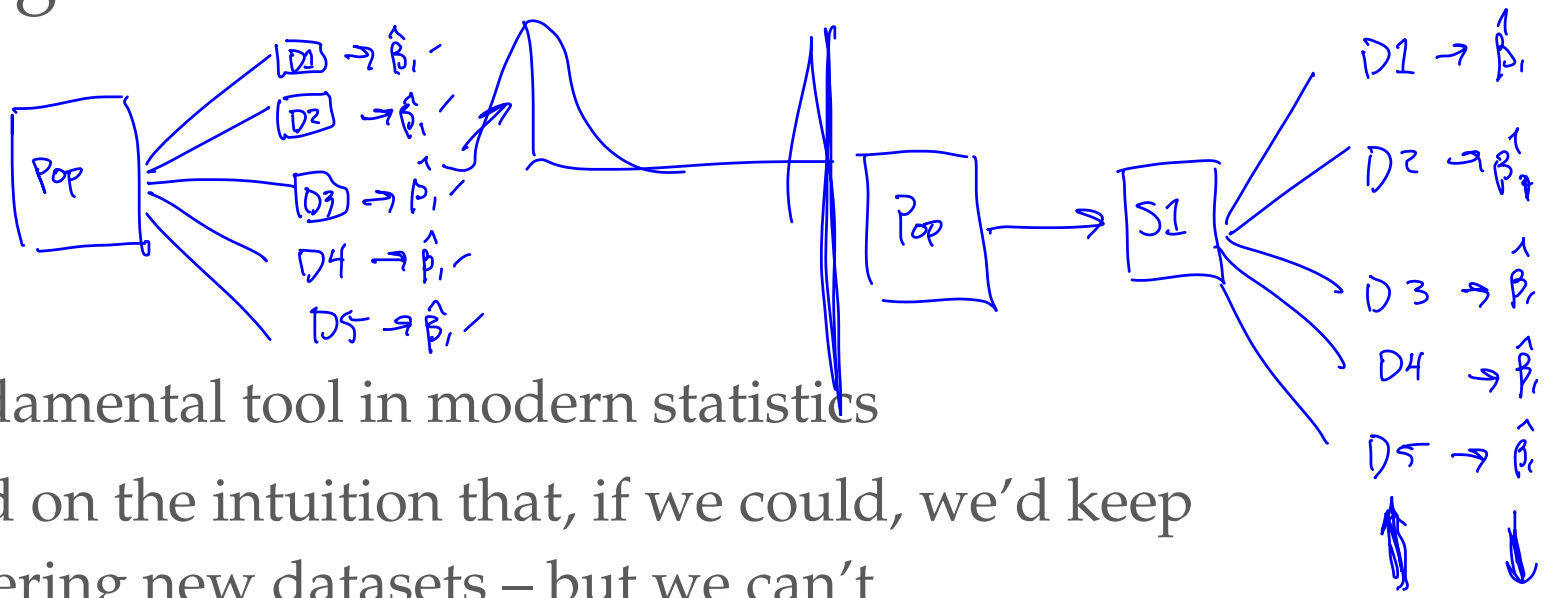
$$\text{Var}(y) = \text{Var}(E(y|x)) + E(\text{Var}(y|x))$$

A prediction interval is given by

$$(\hat{y}|\boldsymbol{x} = \boldsymbol{x}_0) \pm t_{1-\alpha/2,n-p-1}\widehat{se}_{pred}(\hat{y}|\boldsymbol{x}_0)$$

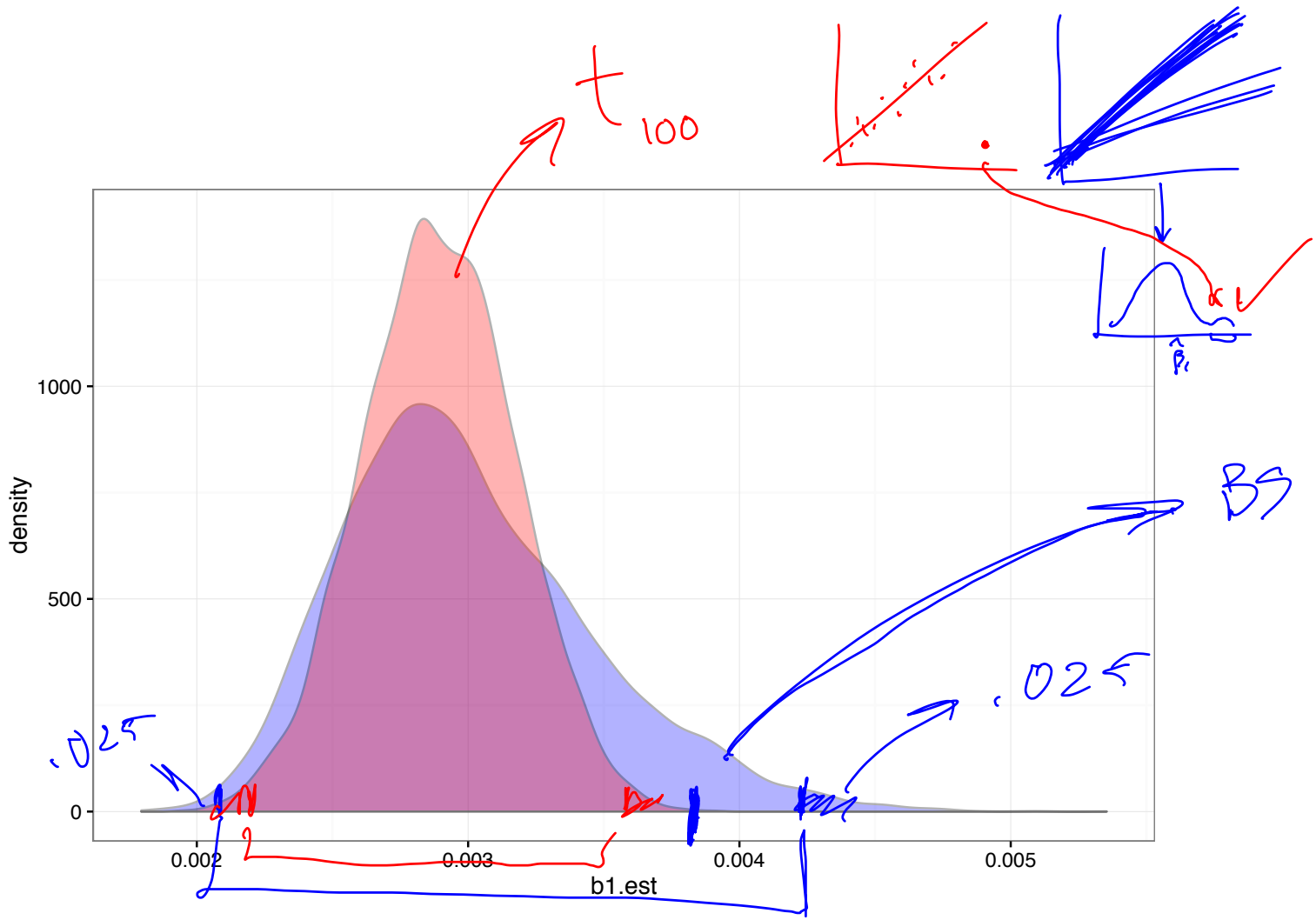
This can be estimated for any $\boldsymbol{x}_0$.

Resampling methods



- Fundamental tool in modern statistics
- Build on the intuition that, if we could, we'd keep gathering new datasets – but we can't
- Use repeated samples of a training set to understand variability
- Computationally intensive ... but we have computers

Bootstrap results



Cross Validation

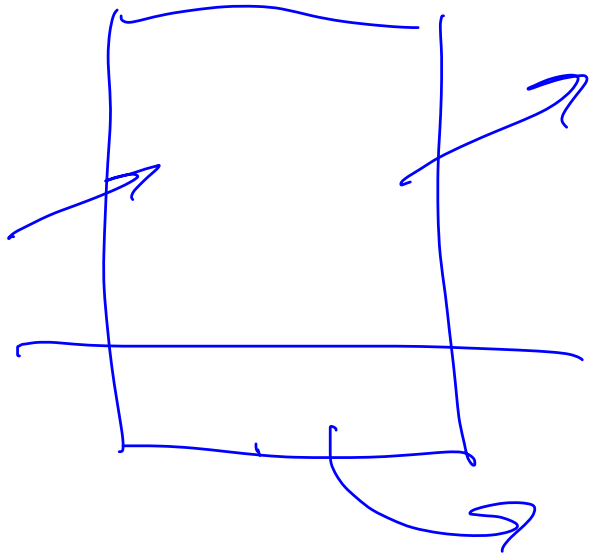
$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$$

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i1}^2 + \epsilon$$

- Focus is on model performance, quantified by prediction error
- We get in-sample performance ...
- But we want generalization to new data
- Most of the time, we don't have an external testing dataset

Cross Validation: one validation set

- Simplest case: create a validation set by randomly splitting the full dataset
- Fit model to training data; compute mean squared prediction error on test set
- Provides way of comparing models (any models ...)



Build model

$\hat{\beta}_1$

$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_2 x_{i2} + \epsilon_i$

$\hat{\beta}_2$

$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\beta}_2 x_{i1}^2 + \epsilon_i$

$X\hat{\beta}_1 = \hat{y}_1$ $\quad \hat{y}_1 - y$ vs

$X\hat{\beta}_2 = \hat{y}_2$ $\quad \hat{y}_2 - y$

Gauss-Markov theorem

Assume the model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $E(\boldsymbol{\epsilon}) = 0$ and $Var(\boldsymbol{\epsilon}) = \sigma^2 I$. Also assume $\boldsymbol{X}$ is a full rank design matrix.

- Among all unbiased linear estimators $\boldsymbol{C}\boldsymbol{y}$ of the regression coefficients $\boldsymbol{\beta}$, the LSE has minimum variance and is unique.

We call the LSEs "BLUE".

Maximum likelihood estimation

Using matrix notation:

$$L(\beta; y) = (2\pi)^{-n/2} (|\sigma^2 I|)^{-1/2} \exp\left\{ -\frac{1}{2} (y - X\beta)^T (\sigma^2 I)^{-1} (y - X\beta) \right\}$$

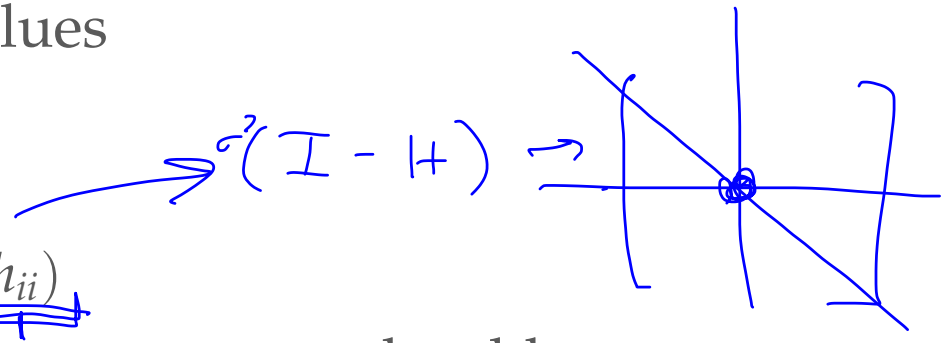
$$\downarrow$$

$$\frac{-1}{2\sigma^2} \underbrace{(y - X\beta)^T (y - X\beta)}_{RSS(\beta)}$$

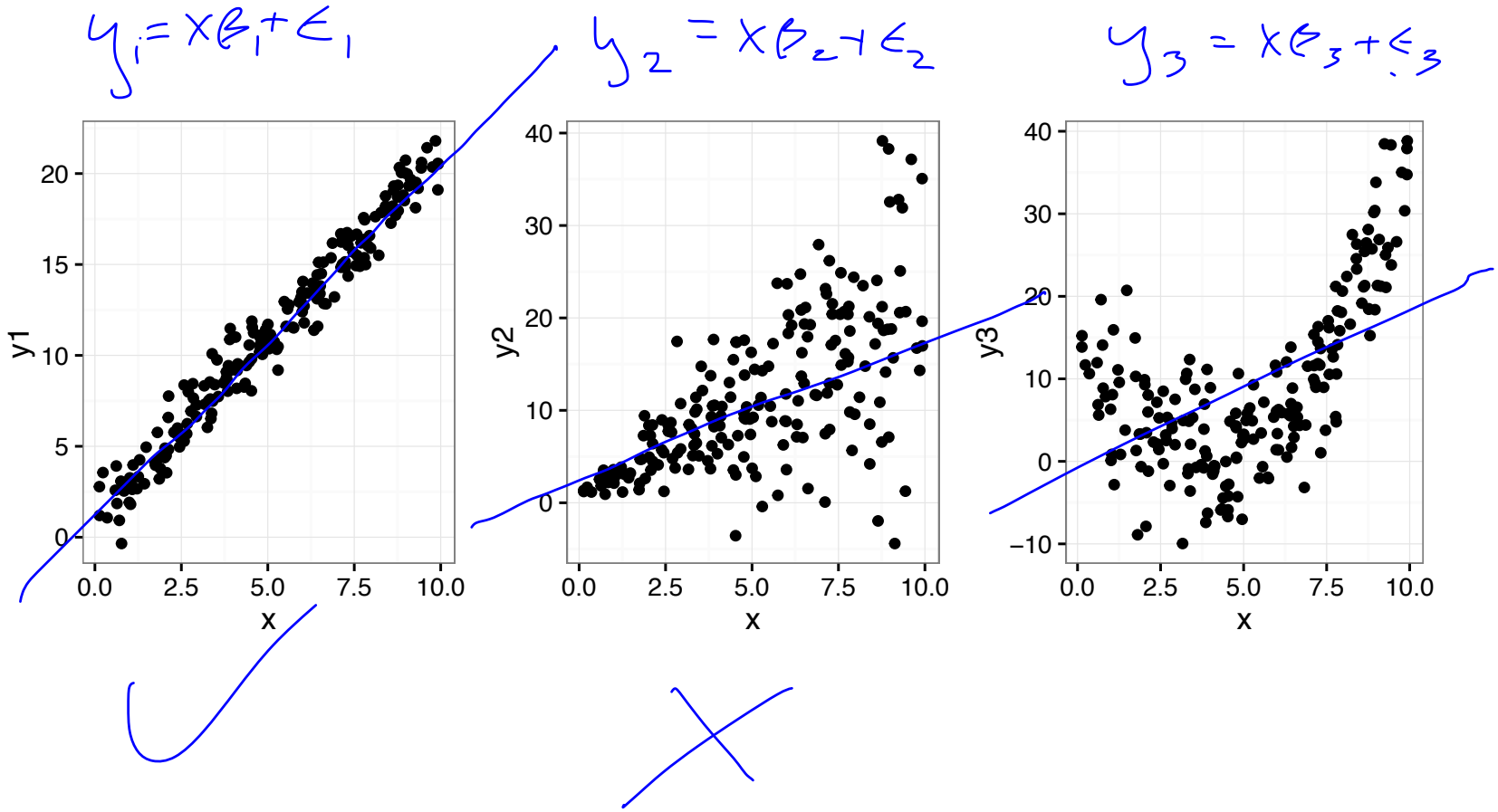
$$(X^T X)^{-1} X^T y$$

Residuals when model is correct

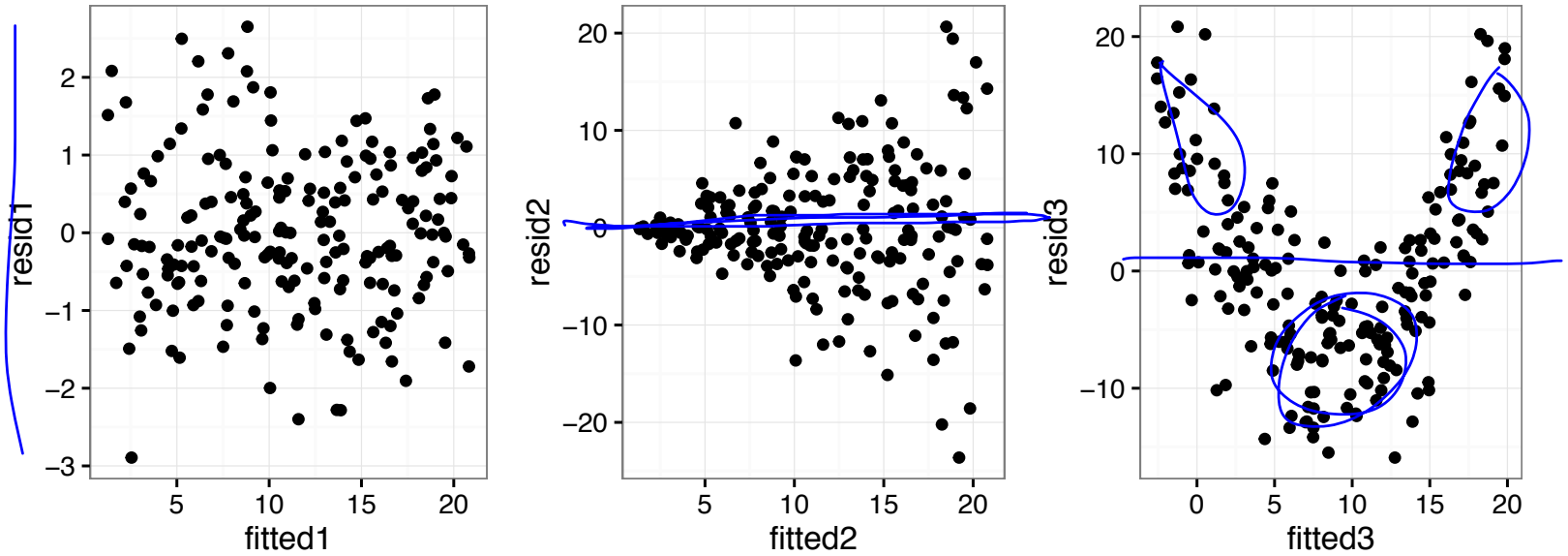
- Often we plot the residuals against one of the predictors or against the fitted values
- What we look for:
  - $E(\hat{\epsilon}|x) = 0$
  - $V(\hat{\epsilon}|x) = \sigma^2(1 - h_{ii})$
- If the model is incorrect, you may be able to spot:
  - Patterns in the residuals
  - Clear non-constant variance



Some data plots



Some residual plots



Model checking

Two major areas of concern:

- Global lack of fit, or general breakdown of model assumptions
  - "Linearity" → $X\beta$ is "Right"
  - Unbiased, uncorrelated errors $E(\epsilon|x) = E(\epsilon) = 0$
  - Constant variance $Var(y|x) = Var(\epsilon|x) = \sigma^2$
  - Independent errors
  - Normality of errors

$\epsilon \sim (0, \sigma^2 I)$

$y \sim (X\beta, \sigma^2 I)$

- Effect of influential points and outliers

Model checking

- Global lack of fit, or general breakdown of model assumptions
  - Residual analysis – QQ plots, residual plots against fitted values and predictors
  - Adjusted variable plots
- Effect of influential points and outliers
  - Measure of leverage, influence, outlying-ness

Variance-stabilizing transformation

Suppose y is strictly positive, $\mu = E(y|x)$, $Var(y|x) = \sigma^2 g(\mu)$

- Replace y with $y^* = T(y)$ such that $Var(y^*|x)$ is approximately constant
- Delta method says $Var(T(y)) = (T'(\mu))^2 \sigma^2 g(\mu)$

$H_0 : e^{\beta} = 0$

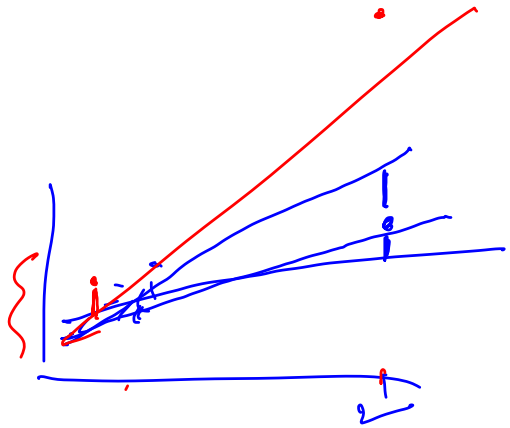
Variance-stabilizing transformation

To get constant variance, we want

$$
\begin{aligned}
(T'(\mu))^2 g(\mu) &= k^2 \quad \text{(constant)} \\
\Rightarrow T'(\mu) &= \frac{k}{\sqrt{g(\mu)}} \\
\Rightarrow T(\mu) &= \int \frac{k}{\sqrt{g(\mu)}} d\mu \\
\Rightarrow T(y) &= \int \frac{k}{\sqrt{g(y)}} dy
\end{aligned}
$$

So the transformation necessary to stabilize the variance really depends on the variance function itself, e.g. $g(\cdot)$

Isolated points



Points can be isolated in three ways

- Leverage point – outlier in x
- Outlier – outlier in $y|x$
- Influential point – a point that largely affects β
  - Deletion influence; $|\hat{\beta} - \hat{\beta}_{(-i)}|$
  - Basically, a high-leverage outlier

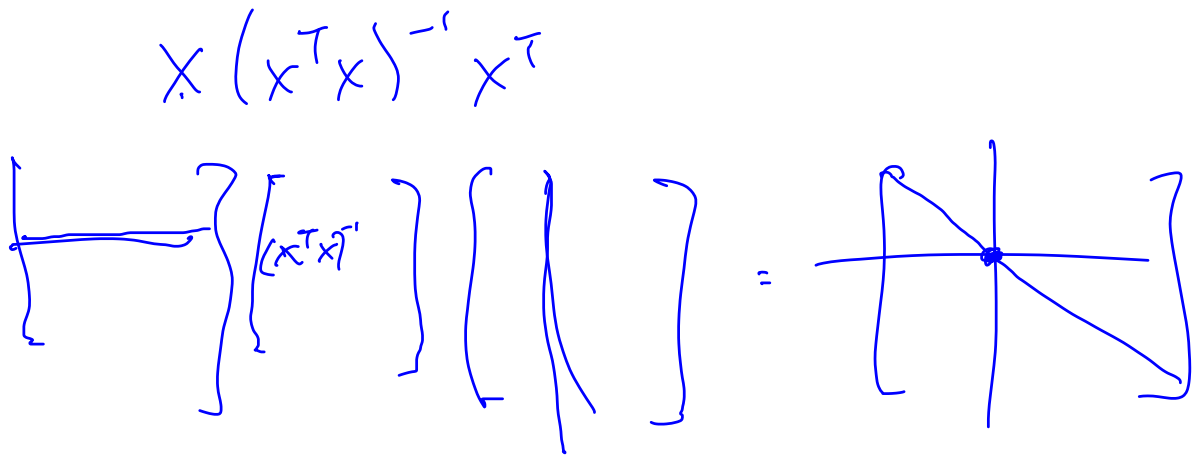
Leverage is measured by the hat matrix, outlying-ness by the residual

Quantifying leverage

We measure leverage (the "distance" of x_i from the distribution of x) using

$$h_{ii} = x_i^T (X^T X)^{-1} x_i$$

where h_{ii} is the $(i,i)^{th}$ entry of the hat matrix.



Leverage

Some notes about the hat matrix

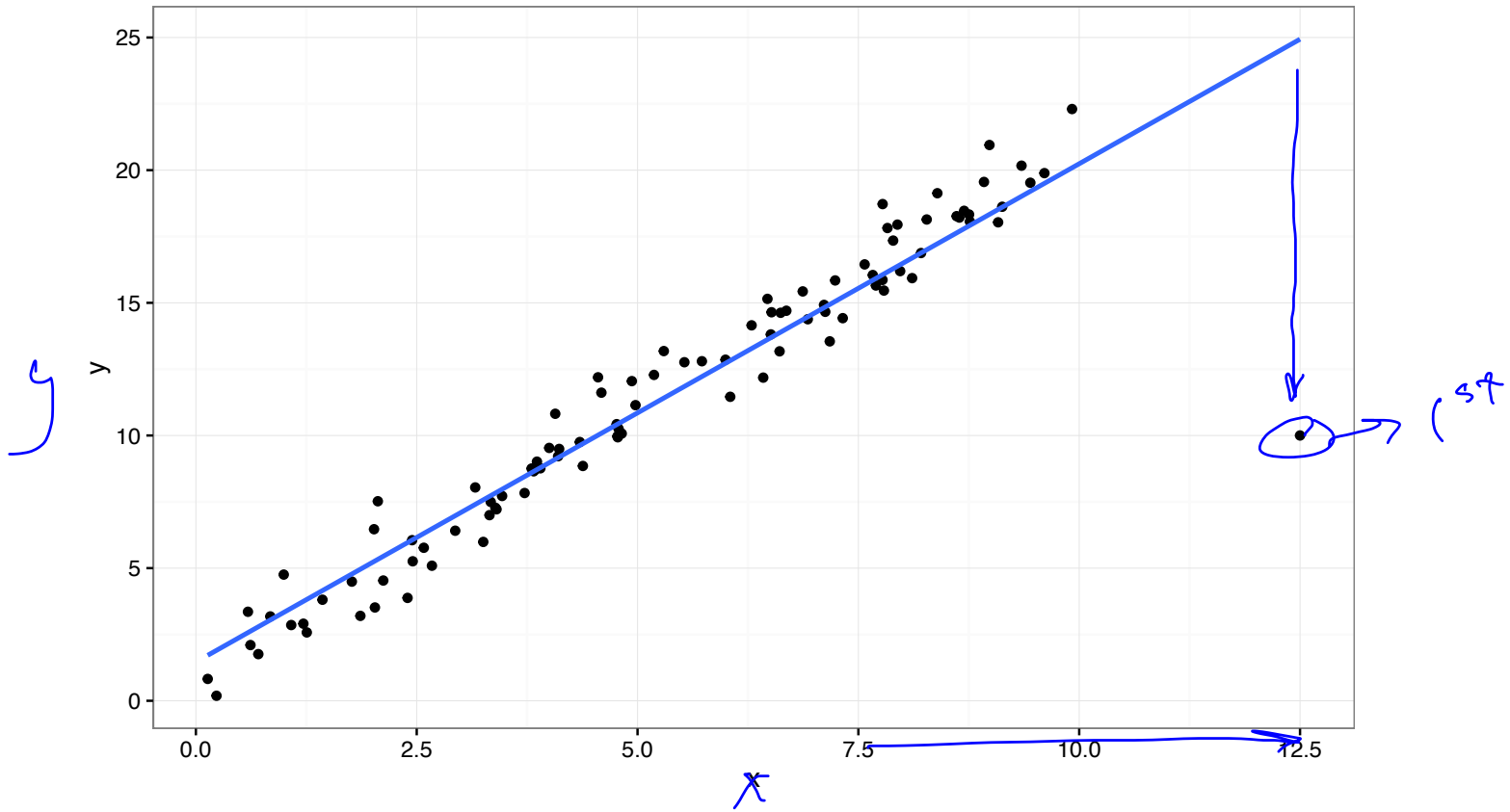
- $\sum_i h_{ii} \overset{def}{=} tr(\boldsymbol{H}) = (p+1)$

$$tr\left(X(X^TX)^{-1}X^T\right) = tr\left(X^TX(X^TX)^{-1}\right) = tr(I)$$

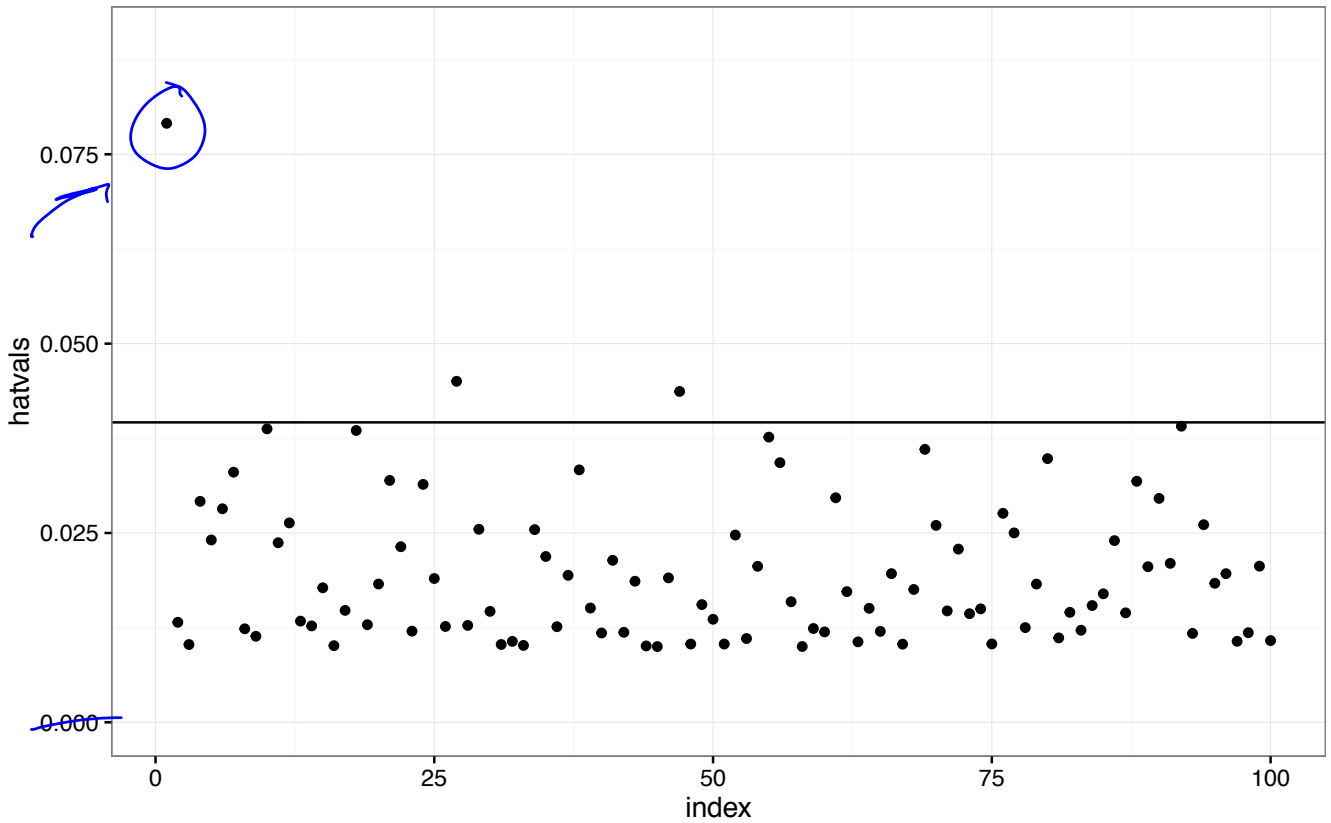
(Note – the trace of the hat matrix generalizes to non-parametric methods, where you don't have a specific number of parameters to count. This is a useful measure of "model size" or "effective degrees of freedom" in these cases.)

$$S_y = \hat{y}$$

Leverage plot



Leverage plot



Outliers

- When we refer to "outliers" we typically mean "points that don't have the same mean structure as the rest of the data"
- Residuals give an idea of "outlying-ness", but we need to standardize somehow
- Remember (from last lecture) $Var(\hat{\epsilon}_i) = \sigma^2(1 - h_{ii})$ ...

Outliers

The *standardized* residual is given by

$$\hat{\epsilon}_i^* = \frac{\hat{\epsilon}_i}{\sqrt{Var(\hat{\epsilon}_i)}} = \frac{\hat{\epsilon}_i}{\hat{\sigma}\sqrt{(1-h_{ii})}}$$

The *Studentized* residual is given by

$$t_i = \frac{\hat{\epsilon}_i}{\hat{\sigma}_{(-i)}\sqrt{(1-h_{ii})}} = \hat{\epsilon}_i^* \left(\frac{n-(p+1)}{n-(p+1)-\hat{\epsilon}_i^{*2}} \right)^{1/2}$$

Studentized residuals follow a $t_{n-(p+1)-1}$ distribution.

Influence

Specifically, deletion influence

- $|\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(-i)}|$ =
- Cook's distance is

$$
\begin{aligned}
D_i &= \frac{(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(-i)})^T (\mathbf{X}^T\mathbf{X})(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(-i)})}{(p+1)\hat{\sigma}^2} \\
&= \frac{(\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}_{(-i)})^T (\hat{\boldsymbol{y}} - \hat{\boldsymbol{y}}_{(-i)})}{(p+1)\hat{\sigma}^2} \\
&= \frac{1}{p+1} \hat{\epsilon}_i^2 \frac{h_{ii}}{1-h_{ii}}
\end{aligned}
$$