Model fit 1

Fit model for earnings as a function of height and generate simulated coefficients.

```
lm.earn <- lm(earn ~ height, heights.clean)
display.xtable(lm.earn)
```

	Estimate	Std. Error	t value
(Intercept)	-61316.28	9525.18	-6.44
height	1262.33	142.11	8.88

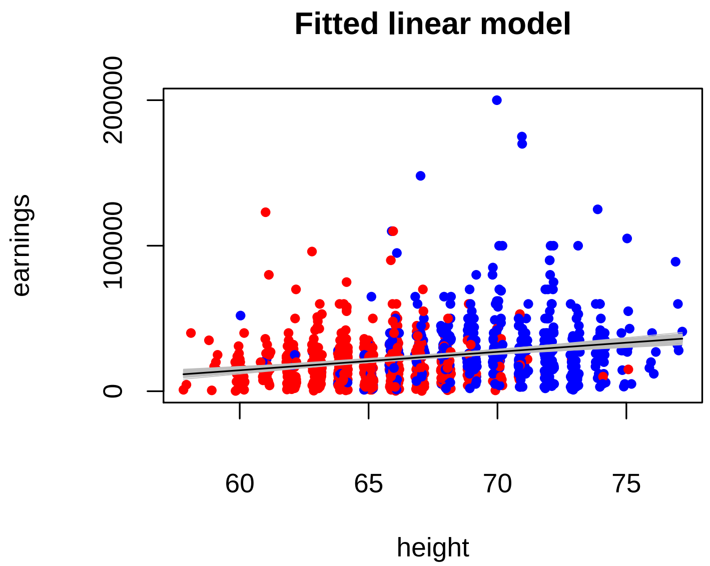
Table: n = 1192 rank = 2 resid sd = 18865.079 R-Squared = 0.062

```
sim.earn <- sim(lm.earn)
beta.hat <- coef(lm.earn)
```


Plot commands

```
## Figure 4.1 (left)
par(mar = c(6, 6, 4, 2) + 0.1)
with(heights.clean, plot(x = height + height.jitter.add, y = earn,
    xlab = "height", ylab = "earnings", pch = 20, mgp = c(4,
        2, 0), yaxt = "n", col = male * 2 + 2, main = "Fitted linear mo
axis(2, c(0, 1e+05, 2e+05), c("0", "100000", "200000"), mgp = c(4,
    1.1, 0))

for (i in 1:100) {
    curve(sim.earn@coef[i, 1] + sim.earn@coef[i, 2] * x, lwd = 0.5,
        col = "gray", add = TRUE)
}
curve(beta.hat[1] + beta.hat[2] * x, add = TRUE, col = "black")
```


Plot model



Grey lines are simulated.

Uncentered

original model

```
fit.4 <- lm(kid_score ~ mom_hs + mom_iq + mom_hs:mom_iq, ki
display.xtable(fit.4)
```

	Estimate	Std. Error	t value
(Intercept)	-11.48	13.76	-0.83
mom_hs	51.27	15.34	3.34
mom_iq	0.97	0.15	6.53
mom_hs:mom_iq	-0.48	0.16	-2.99

Table: n = 434 rank = 4 resid sd = 17.971 R-Squared = 0.23

Note intercept of -11 when Mom's IQ = 0, no high school.

Centered 1
centering by subtracting the mean

```
c_mom_hs <- with(kidiq, mom_hs - mean(mom_hs))
c_mom_iq <- with(kidiq, scale(mom_iq, center = T, scale = F
fit.5 <- lm(kid_score ~ c_mom_hs * c_mom_iq, kidiq)
display.xtable(fit.5)
```

	Estimate	Std. Error	t value
(Intercept)	87.64	0.91	96.57
c_mom_hs	2.84	2.43	1.17
c_mom_iq	0.59	0.06	9.71
c_mom_hs:c_mom_iq	-0.48	0.16	-2.99

Table: n = 434 rank = 4 resid sd = 17.971 R-Squared = 0.23

All but last estimate change. Now does is the "(Intercept)" mean?

Centered 2
using a conventional centering point

```
c2_mom_hs <- with(kidiq, mom_hs - 0.5)
c2_mom_iq <- with(kidiq, mom_iq - 100)
fit.6 <- lm(kid_score ~ c2_mom_hs * c2_mom_iq, kidiq)
display.xtable(fit.6)
```

	Estimate	Std. Error	t value
(Intercept)	86.83	1.21	71.56
c2_mom_hs	2.84	2.43	1.17
c2_mom_iq	0.73	0.08	8.96
c2_mom_hs:c2_mom_iq	-0.48	0.16	-2.99

Table: n = 434 rank = 4 resid sd = 17.971 R-Squared = 0.23

What does the 3rd line of output estimate?

Centered 3
centering by subtracting the mean & dividing by 2 sd

```
z_mom_hs <- with(kidiq, (mom_hs - mean(mom_hs))/(2 * sd(mo
z_mom_iq <- scale(kidiq$mom_iq, TRUE, TRUE)/2  #£
fit.7 <- lm(kid_score ~ z_mom_hs * z_mom_iq, kidiq)
display.xtable(fit.7)
```

	Estimate	Std. Error	t value
(Intercept)	87.64	0.91	96.57
z_mom_hs	2.33	1.99	1.17
z_mom_iq	17.65	1.82	9.71
z_mom_hs:z_mom_iq	-11.94	4.00	-2.99

Table: n = 434 rank = 4 resid sd = 17.971 R-Squared = 0.23

What does not change? Why?

Correlation

In SLR, slope is a function of correlation:

$$\widehat{\beta}_1 = r\frac{\sigma_y}{\sigma_x}$$

What if we standardize x and y?
If we centered?
Note difference between minimizing vertical SSE and minimizing average distance to the line (Principal Components)
Meaning of "regression to the mean"

Log Transform 1

```
log.earn <- log(heights.clean$earn)
earn.logmodel.1 <- lm(log.earn ~ height, heights.clean)
display.xtable(earn.logmodel.1)
```

	Estimate	Std. Error	t value
(Intercept)	5.78	0.45	12.81
height	0.06	0.01	8.74

Table: n = 1192 rank = 2 resid sd = 0.893 R-Squared = 0.06

```
sim.logmodel.1 <- sim(earn.logmodel.1)
beta.hat <- coef(earn.logmodel.1)
```


Log 10

```
log10.earn <- log10(heights.clean$earn)
earn.log10model <- lm(log10.earn ~ height, heights.clean)
display.xtable(earn.log10model)
```

	Estimate	Std. Error	t value
(Intercept)	2.51	0.20	12.81
height	0.03	0.00	8.74

Table: n = 1192 rank = 2 resid sd = 0.388 R-Squared = 0.06

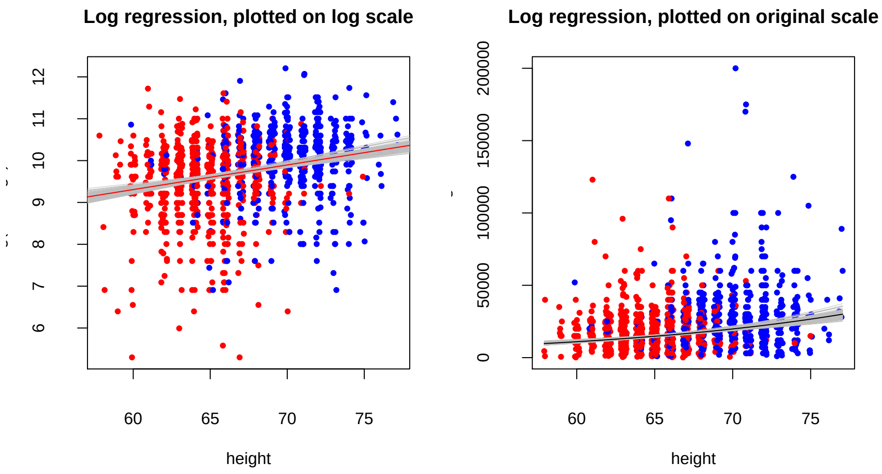
Plotting Log Transform

```
par(mar = c(6, 6, 4, 2) + 0.1)
with(heights.clean, plot(height + runif(n, -0.2, 0.2), log.earn,
    xlab = "height", ylab = "log(earnings)", pch = 20, yaxt = "n",
    mgp = c(4, 2, 0), col = 2 * male + 2, main = "Log regression, plott
axis(2, seq(6, 12, 2), mgp = c(4, 1.1, 0))

for (i in 1:100) curve(sim.logmodel.1@coef[i, 1] + sim.logmodel.1@coef[
    2] * x, lwd = 0.5, col = "gray", add = TRUE)

curve(beta.hat[1] + beta.hat[2] * x, add = TRUE, col = "red")
```


Back-transform Plot

```
par(mar = c(6, 6, 4, 2) + 0.1)
with(heights.clean, plot(height + runif(n, -0.2, 0.2), earn
    xlab = "height", ylab = "earnings", pch = 20, yaxt = "n
    mgp = c(4, 2, 0), col = 2 * male + 2, main = "Log regre
axis(2, c(0, 1e+05, 2e+05), c("0", "100000", "200000"), mgp
    1.1, 0))
for (i in 1:100) curve(exp(sim.logmodel.1@coef[i, 1] + sim
    2] * x), lwd = 0.5, col = "gray", add = TRUE)
curve(exp(beta.hat[1] + beta.hat[2] * x), add = TRUE, col =
```

Figure 4.3



Log Transform 3

Including interactions

```
earn.logmodel.3 <- lm(log.earn ~ height * male, heights.cle
display.xtable(earn.logmodel.3)
```

	Estimate	Std. Error	t value
(Intercept)	8.39	0.84	9.94
height	0.02	0.01	1.30
male	-0.08	1.26	-0.06
height:male	0.01	0.02	0.40

Table: n = 1192 rank = 4 resid sd = 0.881 R-Squared = 0.087

Log Transform 4

Standardized

```
z.height <- with(heights.clean, (height - mean(height))/sd(
earn.logmodel.4 <- lm(log.earn ~ male * z.height, heights.c
display.xtable(earn.logmodel.4)
```

	Estimate	Std. Error	t value
(Intercept)	9.53	0.05	210.88
male	0.42	0.07	5.75
z.height	0.07	0.05	1.30
male:z.height	0.03	0.07	0.40

Table: n = 1192 rank = 4 resid sd = 0.881 R-Squared = 0.087

Log Transform 5

Elasticity

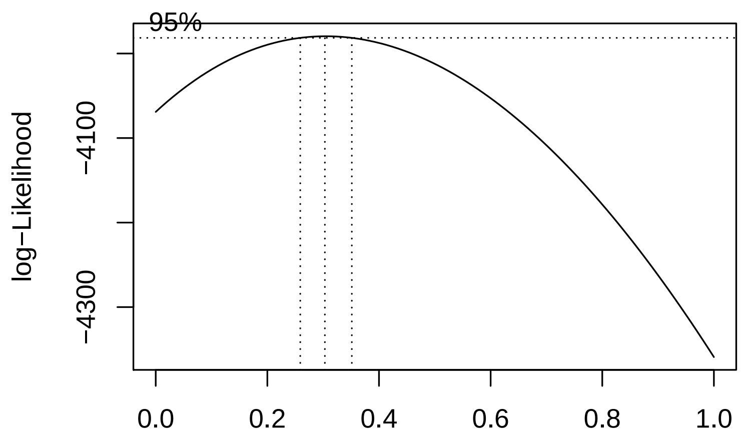
```
log.height <- log(heights.clean$height)
earn.logmodel.5 <- lm(log.earn ~ log.height + male, heights
display.xtable(earn.logmodel.5)
```

	Estimate	Std. Error	t value
(Intercept)	3.62	2.60	1.39
log.height	1.41	0.62	2.26
male	0.42	0.07	5.84

Table: n = 1192 rank = 3 resid sd = 0.881 R-Squared = 0.087

Other Transforms, §4.5

```
MASS::boxcox(lm(earn ~ height + male, heights.clean), lam = seq(0,
    1, 0.1))
```



Indicators

Divide shuttle launches into "cold" ($< 66^o$) or "warm" ($\geq 66^o$) to look at O-ring failures.
Or model failures as a function of temperature?
Is left-handedness a binary variable?
Cut a continuous variable up into bins to make a factor? Or use a smoother?

Identifiability

A model is non-identifiable if some parameters cannot be estimated uniquely (have infinite SE).
Example: a factor with J levels can use J dummy variables, but if the model includes an intercept, we get non-identifiability problem.
Solutions: drop one column, and let this be the reference level.
or drop the intercept (but F tests and R^2 are lost)
or require a constraint like $\sum \tau_i = 0$.
In R `singular.ok = TRUE` allows less than full rank $\mathbf{X}$ without complaint. `NA`'s for missing values.

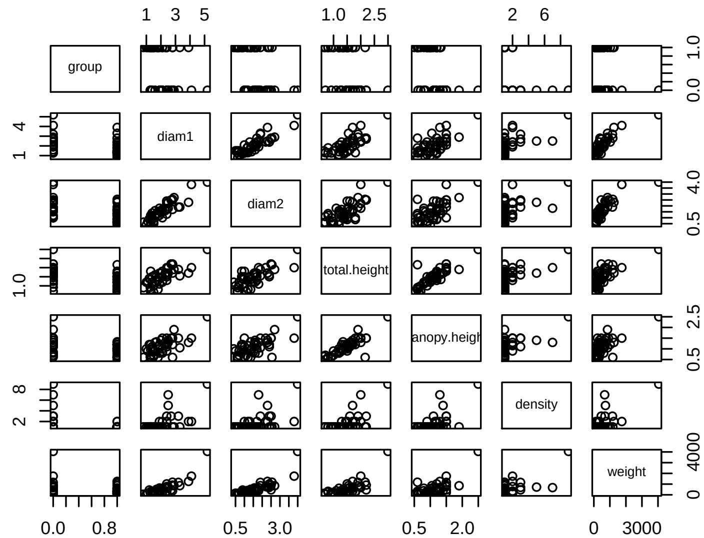
General Principles §4.6

1. Include all "important" predictors
2. Similar predictor variables could be averaged together.
3. Consider interactions when main effects are large.
4. Exclude variables?
   1. No if sign is as expected and p-value is large.
   2. Yes if sign is opposite expected sign and p-value is large.
   3. Maybe if sign is as expected and p-value is small. (Think)
   4. No if sign is as expected and p-value is small.

Mesquite example

```r
mesquite <- read.table("http://www.stat.columbia.edu/~gelman/arm/exampl
    header = TRUE)
names(mesquite)[2:8] <- c("group", "diam1", "diam2", "total.height",
    "canopy.height", "density", "weight")
mesquite$group <- unclass(mesquite$group) - 1  # remove factor
## pairs(mesquite[,-1])
```


Pairs Plot



Data Summary

```r
percentiles <- matrix(sapply(mesquite[, 3:7], quantile, c(0
    0.25, 0.5, 0.75, 0.975)), 5, 5, dimnames = list(percent
    25, 50, 75, 97.5), variable = names(mesquite)[3:7]))

xtable(rbind(percentiles, IQR <- percentiles[4, ] - percent
    ]))
```

	diam1	diam2	total.height	canopy.height	density
2.5	1.01	0.51	0.70	0.60	1.00
25	1.40	1.00	1.20	0.86	1.00
50	1.95	1.52	1.50	1.10	1.00
75	2.48	1.90	1.70	1.30	2.00
97.5	4.07	3.66	2.20	1.85	6.75
	1.08	0.90	0.50	0.44	1.00

Model 1

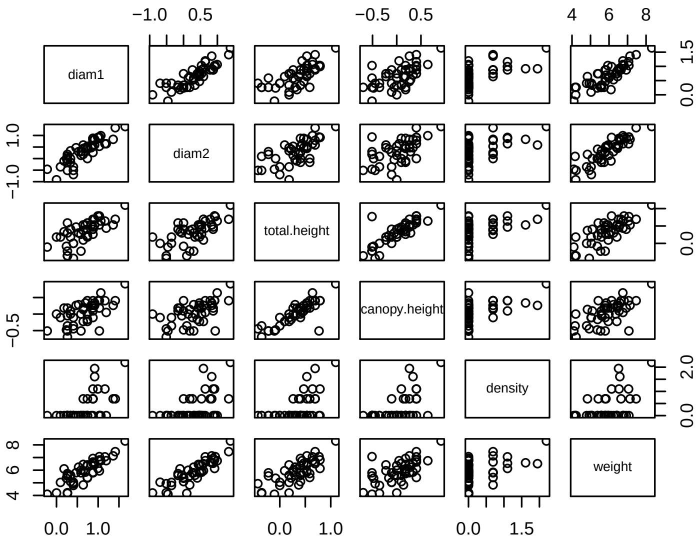
```r
mesq.fit.1 <- lm(weight ~ diam1 + diam2 + canopy.height + t
    density + group, mesquite)
display.xtable(mesq.fit.1)
```

	Estimate	Std. Error	t value
(Intercept)	-1091.89	176.46	-6.19
diam1	189.67	112.76	1.68
diam2	371.46	124.38	2.99
canopy.height	355.67	209.84	1.69
total.height	-101.73	185.57	-0.55
density	131.25	34.36	3.82
group	363.30	100.18	3.63

Table: n = 46 rank = 7 resid sd = 268.96 R-Squared = 0.848

Logs make sense here since weight is related to volume, a product of 3 dimensions.

Log Pairs Plot



Log Model

```
mesq.fit.2 <- lm(log(weight) ~ log(diam1) + log(diam2) + lo
    log(total.height) + log(density) + group, mesquite)
display.xtable(mesq.fit.2)
```

	Estimate	Std. Error	t value
(Intercept)	4.77	0.16	30.75
log(diam1)	0.39	0.28	1.40
log(diam2)	1.15	0.21	5.48
log(canopy.height)	0.37	0.28	1.33
log(total.height)	0.39	0.31	1.26
log(density)	0.11	0.12	0.90
group	0.58	0.13	4.53

Table: n = 46 rank = 7 resid sd = 0.329 R-Squared = 0.887

Volume Model

Total leaf weight is a function of volume of canopy. Build a new variable:

```
canopy.volume <- with(mesquite, diam1 * diam2 * canopy.heig
mesq.fit.3 <- lm(log(weight) ~ log(canopy.volume), mesquite
display.xtable(mesq.fit.3)  # Volume, area & shape model
```

	Estimate	Std. Error	t value
(Intercept)	5.17	0.08	62.07
log(canopy.volume)	0.72	0.05	13.23

Table: n = 46 rank = 2 resid sd = 0.414 R-Squared = 0.799

Can we add to this one? Perhaps surface area and shape?

Volume Model 2

```
canopy.area <- with(mesquite, diam1 * diam2)
canopy.shape <- with(mesquite, diam1/diam2)
mesq.fit.4 <- lm(log(weight) ~ log(canopy.volume) + log(can
    log(canopy.shape) + log(total.height) + log(density) +
    mesquite)
display.xtable(mesq.fit.4)
```

	Estimate	Std. Error	t value
(Intercept)	4.77	0.16	30.75
log(canopy.volume)	0.37	0.28	1.33
log(canopy.area)	0.40	0.29	1.36
log(canopy.shape)	-0.38	0.23	-1.64
log(total.height)	0.39	0.31	1.26
log(density)	0.11	0.12	0.90
group	0.58	0.13	4.53

Table: n = 46 rank = 7 resid sd = 0.329 R-Squared = 0.887

Model 5

```
mesq.fit.5 <- lm(log(weight) ~ log(canopy.volume) + log(car
    group, mesquite)
display.xtable(mesq.fit.5)
```

	Estimate	Std. Error	t value
(Intercept)	4.70	0.12	39.81
log(canopy.volume)	0.61	0.19	3.22
log(canopy.area)	0.29	0.24	1.22
group	0.53	0.12	4.56

Table: n = 46 rank = 4 resid sd = 0.337 R-Squared = 0.873

Model 6

```
mesq.fit.6 <- lm(log(weight) ~ log(canopy.volume) + log(car
    log(canopy.shape) + log(total.height) + group, mesquite
display.xtable(mesq.fit.6)
```

	Estimate	Std. Error	t value
(Intercept)	4.77	0.15	30.84
log(canopy.volume)	0.38	0.28	1.38
log(canopy.area)	0.41	0.29	1.41
log(canopy.shape)	-0.32	0.22	-1.44
log(total.height)	0.42	0.31	1.37
group	0.54	0.12	4.56

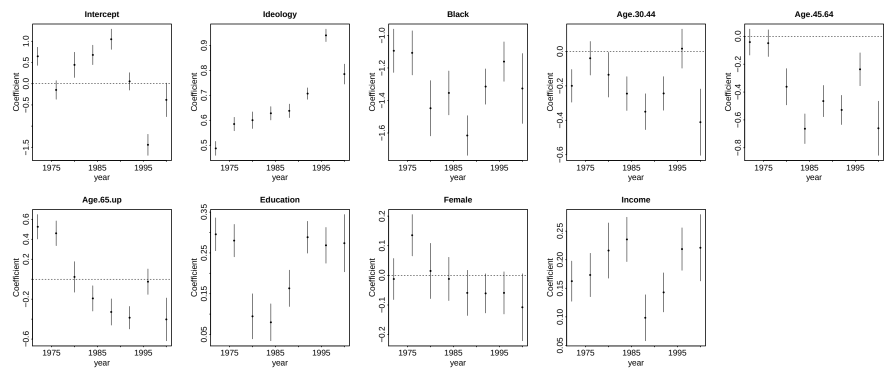
Table: n = 46 rank = 6 resid sd = 0.329 R-Squared = 0.885

Series of models, 4.7

Clean NES elections data to get year, party ID, and nine predictors.

```
regress.year <- function(yr) {
    this.year <- subset(data, nes.year == yr)
    lm.0 <- lm(partyid7 ~ ., data = this.year)
    summary(lm.0)$coef[, 1:2]
}
yrs <- seq(1972, 2000, 4)
yrlyCoef <- array(NA, c(9, 2, 8), dimnames = list(c("Intercept",
    "Ideology", "Black", "Age.30.44", "Age.45.64", "Age.65.up",
    "Education", "Female", "Income"), c("Est", "SE"), yrs))
for (yr in yrs) yrlyCoef[, , (yr - 1968)/4] <- regress.year(yr)
par(mfrow = c(2, 5), mar = c(3, 4, 2, 0))
for (k in 1:9) {
    plot(yrs, yrlyCoef[k, 1, ], pch = 20, cex = 0.5, xlab = "year",
        ylab = "Coefficient", main = dimnames(yrlyCoef)[[1]][k],
        mgp = c(1.2, 0.2, 0), cex.main = 1, cex.axis = 1, cex.lab = 1,
        tcl = -0.1)
    segments(yrs, yrlyCoef[k, 1, ] - 0.67 * yrlyCoef[k, 2, ],
        yrs, yrlyCoef[k, 1, ] + 0.67 * yrlyCoef[k, 2, ], lwd = 0.5)
    abline(h = 0, lwd = 0.5, lty = 2)
}
```


Figure 4.6



Running the same multiple regression in Presidential election years, we can see how some influences on Party identification have changed. Intervals are roughly 50% confidence intervals. Positive coefficients indicate Republican leanings.