

Log Transform 2

```
> earn.logmodel.2 <- lm(log.earn ~ height + male, heights.clean)
> display(earn.logmodel.2)
```

```
lm(formula = log.earn ~ height + male, data = heights.clean)
      coef.est coef.se
(Intercept)  8.15   0.60
height       0.02   0.01
male         0.42   0.07
-----
n = 1192, k = 3
residual sd = 0.88, R-Squared = 0.09
```

Stat 505

Gelman & Hill, Chapter 4

Log Transform 3

Including interactions

```
> earn.logmodel.3 <- lm(log.earn ~ height + male + height:male,
+ heights.clean)
> display(earn.logmodel.3)
```

```
lm(formula = log.earn ~ height + male + height:male, data = heights.clean)
      coef.est coef.se
(Intercept)  8.39   0.84
height       0.02   0.01
male        -0.08   1.26
height:male  0.01   0.02
-----
n = 1192, k = 4
residual sd = 0.88, R-Squared = 0.09
```

Stat 505

Gelman & Hill, Chapter 4

Log Transform 4

Standardized

```
> z.height <- with(heights.clean, (height - mean(height))/sd(height))
> earn.logmodel.4 <- lm(log.earn ~ z.height + male + z.height:male,
+ heights.clean)
> display(earn.logmodel.4)
```

```
lm(formula = log.earn ~ z.height + male + z.height:male, data = heights.clean)
      coef.est coef.se
(Intercept)  9.53   0.05
z.height     0.07   0.05
male         0.42   0.07
z.height:male 0.03   0.07
-----
n = 1192, k = 4
residual sd = 0.88, R-Squared = 0.09
```

Stat 505

Gelman & Hill, Chapter 4

Log Transform 5

Elasticity

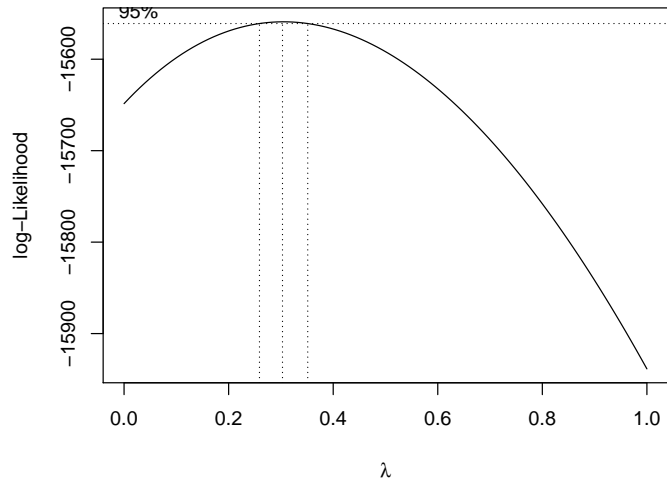
```
> log.height <- log(heights.clean$height)
> earn.logmodel.5 <- lm(log.earn ~ log.height + male, heights.clean)
> display(earn.logmodel.5)
```

```
lm(formula = log.earn ~ log.height + male, data = heights.clean)
      coef.est coef.se
(Intercept)  3.62   2.60
log.height   1.41   0.62
male         0.42   0.07
-----
n = 1192, k = 3
residual sd = 0.88, R-Squared = 0.09
```

Stat 505

Gelman & Hill, Chapter 4

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



Divide shuttle launches into "cold" ($< 66^\circ$) or "warm" ($\geq 66^\circ$) 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?

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.

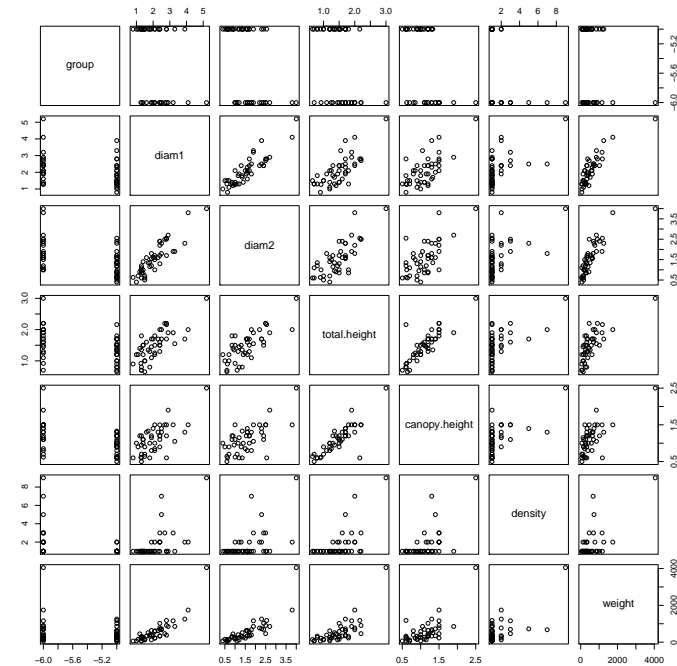
or drop the intercept (but F tests and R^2 are lost)

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.

- 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.

```
> names(mesquite)[2:8] <- c("group", "diam1", "diam2", "total.height",
+ "canopy.height", "density", "weight")
> mesquite$group <- unclass(mesquite$group) - 1
> pairs(mesquite[, -1])
```



```
> summary(mesquite[, -1])[c(1, 3, 6), ]
```

group	diam1	diam2	total.height
Min. : -6.000	Min. : 0.800	Min. : 0.400	Min. : 0.650
Median : -5.000	Median : 1.950	Median : 1.525	Median : 1.500
Max. : -5.000	Max. : 5.200	Max. : 4.000	Max. : 3.000
canopy.height	density	weight	
Min. : 0.5000	Min. : 1.000	Min. : 60.2	
Median : 1.1000	Median : 1.000	Median : 361.9	
Max. : 2.5000	Max. : 9.000	Max. : 4052.0	

```
> apply(mesquite[, 2:7], 2, IQR)
```

group	diam1	diam2	total.height	canopy.height	density
1.000	1.075	0.900	0.5000	0.4375	1.0s

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

```
lm(formula = weight ~ diam1 + diam2 + canopy.height + total.height +
    density + group, data = mesquite)
            coef.est coef.se
(Intercept)  1087.88   524.74
diam1         189.67   112.76
diam2         371.46   124.38
canopy.height  355.67   209.84
total.height  -101.73   185.57
density        131.25    34.36
group         363.30   100.18

---
n = 46, k = 7
residual sd = 268.96, R-Squared = 0.85
```

Log Model

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

```
lm(formula = log(weight) ~ log(diam1) + log(diam2) + log(total.height) + log(density) + group, data = mesquite)
```

	coef.est	coef.se
(Intercept)	8.27	0.74
log(diam1)	0.39	0.28
log(diam2)	1.15	0.21
log(canopy.height)	0.37	0.28
log(total.height)	0.39	0.31
log(density)	0.11	0.12
group	0.58	0.13

```
n = 46, k = 7
residual sd = 0.33, R-Squared = 0.89
```

Stat 505

Gelman & Hill, Chapter 4

Volume Model

```
> canopy.volume <- with(mesquite, diam1 * diam2 * canopy.height)
> mesq.fit.3 <- lm(log(weight) ~ log(canopy.volume), mesquite)
> display(mesq.fit.3)
```

```
lm(formula = log(weight) ~ log(canopy.volume), data = mesquite)
```

	coef.est	coef.se
(Intercept)	5.17	0.08
log(canopy.volume)	0.72	0.05

```
n = 46, k = 2
residual sd = 0.41, R-Squared = 0.80
```

Stat 505

Gelman & Hill, Chapter 4

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(canopy.area) + log(canopy.shape) + log(total.height) + log(density) + group, mesquite)
> display(mesq.fit.4)
```

```
lm(formula = log(weight) ~ log(canopy.volume) + log(canopy.area) + log(canopy.shape) + log(total.height) + log(density) + group, data = mesquite)
```

	coef.est	coef.se
(Intercept)	8.27	0.74
log(canopy.volume)	0.37	0.28
log(canopy.area)	0.40	0.29
log(canopy.shape)	-0.38	0.23
log(total.height)	0.39	0.31
log(density)	0.11	0.12
group	0.58	0.13

```
n = 46, k = 7
residual sd = 0.33, R-Squared = 0.89
```

Stat 505

Gelman & Hill, Chapter 4

Model 5

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

```
lm(formula = log(weight) ~ log(canopy.volume) + log(canopy.area) + log(canopy.shape) + group, data = mesquite)
```

	coef.est	coef.se
(Intercept)	7.86	0.61
log(canopy.volume)	0.61	0.19
log(canopy.area)	0.29	0.24
group	0.53	0.12

```
n = 46, k = 4
residual sd = 0.34, R-Squared = 0.87
```

Stat 505

Gelman & Hill, Chapter 4

Model 6

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

```
lm(formula = log(weight) ~ log(canopy.volume) + log(canopy.area) +  
log(canopy.shape) + log(total.height) + group, data = mesquite)
```

	coef.est	coef.se
(Intercept)	8.00	0.68
log(canopy.volume)	0.38	0.28
log(canopy.area)	0.41	0.29
log(canopy.shape)	-0.32	0.22
log(total.height)	0.42	0.31
group	0.54	0.12

```
n = 46, k = 6  
residual sd = 0.33, R-Squared = 0.88
```