

Another Generalized Inverse

```
> summary(warp.fit ← lm(breaks ~ tension, warpbreaks))$coef
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	36	2.8	13.0	8.3e-18
tensionM	-10	4.0	-2.5	1.5e-02
tensionH	-15	4.0	-3.7	5.0e-04

```
> coef(warp.fit2 ← lm(breaks ~ tension -1, warpbreaks))
```

tensionL	tensionM	tensionH
36	26	22

```
> Lambda ← matrix(c(-1,1,0, -1,0,1, 0,1,-1),byrow=TRUE,3,3)
> as.numeric(Lambda %*% coef(warp.fit2))
```

```
[1] -10.0 -14.7 4.7
```

```
> sqrt(diag(Lambda %*% summary(warp.fit2)$cov.unscaled %*%
+ t(Lambda))) * summary(warp.fit2)$sigma
```

```
[1] 4 4 4
```

Stat 505

Gelman & Hill, Chapter 3

Residuals

$$r_i = y_i - X_i \hat{\beta} \quad \epsilon = \mathbf{y} - \mathbf{X} \hat{\beta}$$

$s^2 = \hat{\sigma}^2 = \epsilon^T \epsilon / (n - k) = \sum r_i^2 / (n - k)$ where n is the number of rows and k is the rank of \mathbf{X} .

The sampling distribution of $\frac{\hat{\sigma}^2}{\sigma^2}$ is χ_{n-k}^2 .

$R^2 = 1 - \frac{\hat{\sigma}^2}{s_y^2}$ is proportion of variance of \mathbf{y} explained by the model.

The model must contain an intercept.

p 42 Do the authors drop predictors with coefficients close to 0 (in SE's)? Stay tuned for §4.6

Stat 505

Gelman & Hill, Chapter 3

Compare the fits

See how the two var-cov matrices compare:

```
> xtable(summary(warp.fit)$cov.unscaled * summary(warp.fit)$sigma
```

	(Intercept)	tensionM	tensionH
(Intercept)	7.84	-7.84	-7.84
tensionM	-7.84	15.68	7.84
tensionH	-7.84	7.84	15.68

```
> xtable(summary(warp.fit2)$cov.unscaled * summary(warp.fit2)$sigma
```

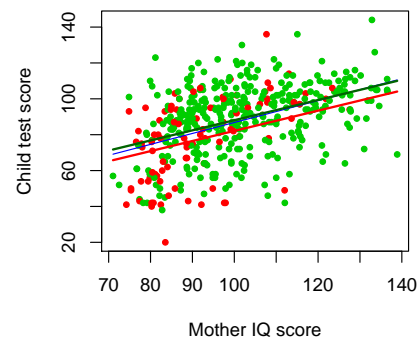
	tensionL	tensionM	tensionH
tensionL	7.84	-0.00	-0.00
tensionM	-0.00	7.84	-0.00
tensionH	-0.00	-0.00	7.84

Stat 505

Gelman & Hill, Chapter 3

Drawing Lines

```
> kidfit.2 ← lm(kid.score ~ mom.iq, kidiq)
> plot(kid.score ~ mom.iq, data=kidiq, xlab="Mother IQ score",
+      ylab="Child test score", col=mom.hs+2, pch=20)
> curve(coef(kidfit.2)[1] + coef(kidfit.2)[2]*x, add=TRUE, col="black")
> fit.3 ← lm(kid.score ~ mom.hs + mom.iq, kidiq)
> curve(cbind(1, 1, x) %*% coef(fit.3), add=TRUE, col="darkred")
> curve(cbind(1, 0, x) %*% coef(fit.3), add=TRUE, col="red")
```

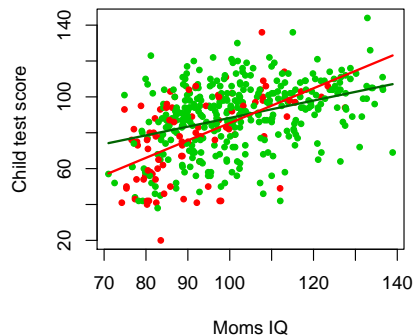


Stat 505

Gelman & Hill, Chapter 3

Interaction Lines

```
> kidfit.4 <- lm(kid.score ~ mom.hs * mom.iq, kidiq)
> plot(kid.score ~ mom.iq, data=kidiq, xlab="Moms IQ",
+      ylab="Child test score", col=mom.hs+2, pch=20)
> curve(cbind(1, 1, x, 1*x) %>% coef(kidfit.4), add=TRUE,
+       col="darkgreen", lwd=2)
> curve(cbind(1, 0, x, 0*x) %>% coef(kidfit.4), add=TRUE,
+       col="red", lwd=2)
```



Stat 505

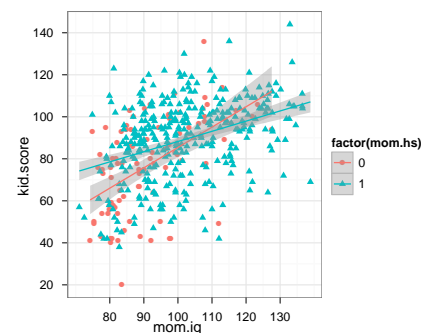
Gelman & Hill, Chapter 3

Uncertainty

How well is the line fit?

For each row of data, $SE(\hat{y}_i) = s\sqrt{\mathbf{X}_i\mathbf{V}_\beta\mathbf{X}_i^T}$

```
> require(ggplot2)
> myplot <- qplot(mom.iq, kid.score, data=kidiq, shape=fac
+               colour = factor(mom.hs) ) +theme_bw()
> print(myplot + geom_smooth(method = "lm"))
```



Stat 505

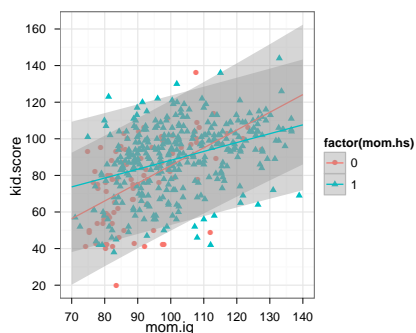
Gelman & Hill, Chapter 3

Uncertainty 2

How well do we predict new points?

A new point has extra variance ($\hat{\sigma}^2$ so now the predictive error is $\sqrt{1 + \mathbf{X}_i\mathbf{V}_\beta\mathbf{X}_i^T}$

```
> predictDF <- data.frame(mom.iq=rep(7:14*10,2), mom.hs=rep(0,14)+rep(1,14))
> predictDF <- cbind(predictDF, predict(kidfit.4, newdata=predictDF))
> names(predictDF)[3] <- "kid.score"
> print(myplot + geom_smooth(aes(ymin = lwr, ymax = upr)))
```



Stat 505

Gelman & Hill, Chapter 3

The Usual Assumptions

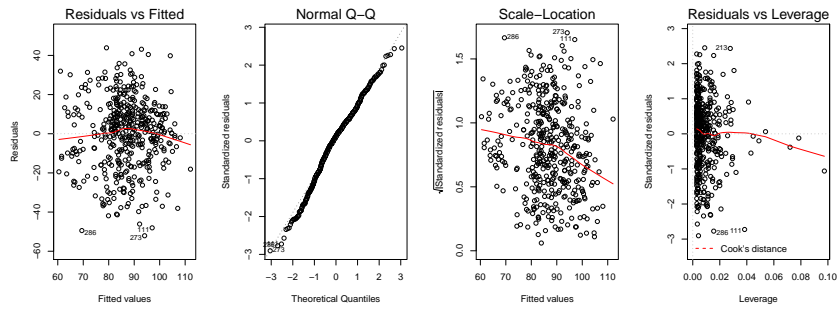
- 1 Data are valid – will answer our question.
- 2 Model is valid – properly specified.
- 3 Independent errors.
- 4 Constant variance = homo– not hetero–scedastic
- 5 Normality of errors. Never check raw \mathbf{y} for normality.

Stat 505

Gelman & Hill, Chapter 3

Diagnostic plots

```
> par(mfrow=c(1,4))  
> plot(kidfit.4)
```



Check validity with new data. (extrapolate?)

Data used to create the estimated model always fits it better than data which got no “say” in estimation.

Later: “Can the model generate data like the data we see?”