Notes 5: Multiple Linear Regression

Nathaniel E. Helwig

Department of Statistics
University of Illinois at Urbana-Champaign

Stat 420: Methods of Applied Statistics
Section N1U/N1G – Spring 2014

Outline of Notes

1) Intro to MLR Model:

- Model form (scalar)
- MLR assumptions
- Model form (matrix)

2) Estimation of MLR Model:

- Ordinary least squares
- Calculus derivation
- Maximum likelihood

3) Inferences in MLR:

- Estimating error variance
- Distribution of estimator
- Single slope tests
- Multiple slopes tests
- Linear combinations
- CIs, PIs, and CRs
- Example: GPA

MLR Model: Form

The multiple linear regression model has the form

$$y_i = b_0 + \sum_{j=1}^{p} b_j x_{ij} + e_i$$

for $i \in \{1, \ldots, n\}$ where

- $y_i \in \mathbb{R}$ is the real-valued response for the i-th observation
- $b_0 \in \mathbb{R}$ is the regression intercept
- $b_j \in \mathbb{R}$ is the j-th predictor's regression slope
- $x_{ij} \in \mathbb{R}$ is the j-th predictor for the i-th observation
- $e_i \stackrel{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$ is Gaussian measurement error

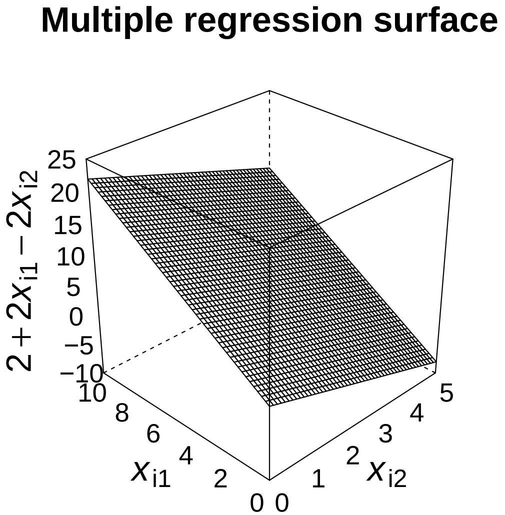
MLR Model: Name

The model is *multiple* because we have $p > 1$ predictors.

The model is *linear* because y_i is a linear function of the parameters (b_0, b_1, ..., b_p are the parameters).

The model is a *regression* model because we are modeling a response variable (Y) as a function of predictor variables ($X_1, \ldots, X_p$).

MLR Model: Visualization



MLR Model: Visualization (R code)

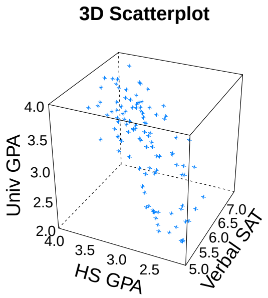
```
> library(lattice)
> x11(height=6,width=6)
> x1=seq(0,10,length.out=50)
> x2=seq(0,5,length.out=50)
> mydata=expand.grid(x1,x2)
> y=2+2*mydata[,1]-2*mydata[,2]
> wireframe(y~mydata[,2]*mydata[,1],xlab=list(label=expression(italic(x)[i2]),cex=2),
+          ylab=list(label=expression(italic(x)[i1]),cex=2),
+          zlab=list(label=expression(2+2*italic(x)[i1]-2*italic(x)[i2]),cex=2,rot=90,vjust=0),
+          scales=list(arrows=FALSE,cex=1.5),
+          main=list(label="Multiple regression surface",cex=2,vjust=2),
+          zlim=c(-10,25),screen=list(z=45,x=-60),
+          par.settings=list(axis.line=list(col="transparent")))
```

MLR Model: Example

Predict university GPA from high school GPA and SAT verbal scores.

Multiple linear regression equation for modeling university GPA:

$$(\mathrm{U_{gpa}})_i = 0.6839 + 0.5628(\mathrm{H_{gpa}})_i + 0.1265(\mathrm{SAT_{verb}}/100)_i + (\mathrm{error})_i$$



Data from http://onlinestatbook.com/2/regression/intro.html

MLR Assumptions: Overview

The fundamental assumptions of the MLR model are:

1. Relationship between x_j and y is linear (given other predictors)
2. x_{ij} and y_i are observed random variables (constants)
3. $e_i \stackrel{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$ is an unobserved random variable
4. $b_0, b_1, \ldots, b_p$ are unknown constants
5. $(y_i | x_{i1}, \ldots, x_{ip}) \stackrel{\text{ind}}{\sim} \mathrm{N}(b_0 + \sum_{j=1}^{p} b_j x_{ij}, \sigma^2)$
   note: homogeneity of variance

Note: b_j is expected increase in Y for 1-unit increase in X_j with all other predictor variables held constant

MLR Model: Form (revisited)

The multiple linear regression model has the form

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e}$$

where

- $\mathbf{y} = (y_1, \ldots, y_n)' \in \mathbb{R}^n$ is the $n \times 1$ response vector
- $\mathbf{X} = [\mathbf{1}_n, \mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n \times (p+1)}$ is the $n \times (p+1)$ design matrix
  - $\mathbf{1}_n$ is an $n \times 1$ vector of ones
  - $\mathbf{x}_j = (x_{1j}, \ldots, x_{nj})' \in \mathbb{R}^n$ is j-th predictor vector ($n \times 1$)
- $\mathbf{b} = (b_0, b_1, \ldots, b_p)' \in \mathbb{R}^{p+1}$ is $(p+1) \times 1$ vector of coefficients
- $\mathbf{e} = (e_1, \ldots, e_n)' \in \mathbb{R}^n$ is the $n \times 1$ error vector

MLR Model: Assumptions (revisited)

In matrix terms, the error vector is multivariate normal:

$$\mathbf{e} \sim \mathrm{N}(\mathbf{0}_n, \sigma^2 \mathbf{I}_n)$$

In matrix terms, the response vector is multivariate normal given $\mathbf{X}$:

$$(\mathbf{y}|\mathbf{X}) \sim \mathrm{N}(\mathbf{X}\mathbf{b}, \sigma^2 \mathbf{I}_n)$$

Ordinary Least Squares: Matrix Form

The ordinary least squares (OLS) problem is

$$\min_{\mathbf{b}\in\mathbb{R}^{p+1}} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2$$

where $\|\cdot\|$ denotes the Frobenius norm.

The OLS solution has the form

$$\hat{\mathbf{b}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

which is the same formula from SLR!

Fitted Values and Residuals

SCALAR FORM:

Fitted values are given by

$$\hat{y}_i = \hat{b}_0 + \sum_{j=1}^{p} \hat{b}_j x_{ij}$$

and *residuals* are given by

$$\hat{e}_i = y_i - \hat{y}_i$$

MATRIX FORM:

Fitted values are given by

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\mathbf{b}}$$

and *residuals* are given by

$$\hat{\mathbf{e}} = \mathbf{y} - \hat{\mathbf{y}}$$

Hat Matrix (same as SLR model)

Note that we can write the fitted values as

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{X}\hat{\mathbf{b}} \\ &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\ &= \mathbf{H}\mathbf{y}\end{aligned}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the *hat matrix*.

$\mathbf{H}$ is a symmetric and idempotent matrix: $\mathbf{H}\mathbf{H} = \mathbf{H}$

$\mathbf{H}$ projects $\mathbf{y}$ onto the column space of $\mathbf{X}$.

Example #1: Used Car Data

Suppose we have the following data from a random sample of $n = 8$ car sales at Bob's Used Car's lot:

Selling price ($1000s): y	11	15	13	14	0	19	16	8
Hours of required work: x_1	0	11	11	7	4	10	5	8
Buying price ($1000s): x_2	1	5	4	3	1	4	4	2

Bob thinks that he can predict a car's selling price (y) from the number of work hours the car requires (x_1) and the price he pays for it (x_2).

Assume the multiple linear regression model: $y_i = b_0 + \sum_{j=1}^{2} b_j x_{ij} + e_i$ with $e_i \overset{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$. Find the least-squares regression line.

Example #1: OLS Estimation

The necessary crossproduct statistics are given by

$$\mathbf{X}'\mathbf{X} = \begin{pmatrix} 8 & 56 & 24 \\ 56 & 496 & 200 \\ 24 & 200 & 88 \end{pmatrix} \qquad \mathbf{X}'\mathbf{y} = \begin{pmatrix} 96 \\ 740 \\ 336 \end{pmatrix}$$

$$(\mathbf{X}'\mathbf{X})^{-1} = \begin{pmatrix} 0.7125 & -0.025 & -0.1375 \\ -0.025 & 0.025 & -0.05 \\ -0.1375 & -0.05 & 0.1625 \end{pmatrix}$$

so the least-squares regression coefficients are

$$\hat{\mathbf{b}} = \begin{pmatrix} \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} 0.7125 & -0.025 & -0.1375 \\ -0.025 & 0.025 & -0.05 \\ -0.1375 & -0.05 & 0.1625 \end{pmatrix} \begin{pmatrix} 96 \\ 740 \\ 336 \end{pmatrix} = \begin{pmatrix} 3.7 \\ -0.7 \\ 4.4 \end{pmatrix}$$

Regression Sums-of-Squares: Scalar Form

In MLR models, the relevant sums-of-squares are

- *Sum-of-Squares Total*: $SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$
- *Sum-of-Squares Regression*: $SSR = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$
- *Sum-of-Squares Error*: $SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

The corresponding degrees of freedom are

- SST: $df_T = n - 1$
- SSR: $df_R = p$
- SSE: $df_E = n - p - 1$

Regression Sums-of-Squares: Matrix Form

In MLR models, the relevant sums-of-squares are

$$
\begin{aligned}
SST &= \sum_{i=1}^{n} (y_i - \bar{y})^2 \\
&= \mathbf{y}' \left[\mathbf{I}_n - (1/n)\mathbf{J} \right] \mathbf{y} \\
SSR &= \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 \\
&= \mathbf{y}' \left[\mathbf{H} - (1/n)\mathbf{J} \right] \mathbf{y} \\
SSE &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\
&= \mathbf{y}' \left[\mathbf{I}_n - \mathbf{H} \right] \mathbf{y}
\end{aligned}
$$

Note: $\mathbf{J}$ is an $n \times n$ matrix of ones

Partitioning the Variance (same as SLR model)

We can partition the total variation in y_i as

$$
\begin{aligned}
SST &= \sum_{i=1}^{n}(y_i - \bar{y})^2 \\
&= \sum_{i=1}^{n}(y_i - \hat{y}_i + \hat{y}_i - \bar{y})^2 \\
&= \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + 2\sum_{i=1}^{n}(\hat{y}_i - \bar{y})(y_i - \hat{y}_i) \\
&= SSR + SSE + 2\sum_{i=1}^{n}(\hat{y}_i - \bar{y})\hat{e}_i \\
&= SSR + SSE
\end{aligned}
$$

Partitioning the Variance: Proof (same as SLR model)

To show that $\sum_{i=1}^{n}(\hat{y}_i - \bar{y})\hat{e}_i = 0$, note that

$$\begin{aligned}
\sum_{i=1}^{n}(\hat{y}_i - \bar{y})\hat{e}_i &= (\mathbf{Hy} - n^{-1}\mathbf{1}_n\mathbf{1}_n'\mathbf{y})'(\mathbf{y} - \mathbf{Hy}) \\
&= \mathbf{y}'\mathbf{Hy} - \mathbf{y}'\mathbf{H}^2\mathbf{y} - n^{-1}\mathbf{y}'\mathbf{1}_n\mathbf{1}_n'\mathbf{y} + n^{-1}\mathbf{y}'\mathbf{1}_n\mathbf{1}_n'\mathbf{Hy} \\
&= \mathbf{y}'\mathbf{Hy} - \mathbf{y}'\mathbf{H}^2\mathbf{y} - n^{-1}\mathbf{y}'\mathbf{1}_n\mathbf{1}_n'\mathbf{y} + n^{-1}\mathbf{y}'\mathbf{H}\mathbf{1}_n\mathbf{1}_n'\mathbf{y} \\
&= 0
\end{aligned}$$

given that $\mathbf{H}^2 = \mathbf{H}$ (because $\mathbf{H}$ is idempotent) and $\mathbf{H}\mathbf{1}_n\mathbf{1}_n' = \mathbf{1}_n\mathbf{1}_n'$ (because $\mathbf{1}_n\mathbf{1}_n'$ is within the column space of $\mathbf{X}$ and $\mathbf{H}$ is the projection matrix for the column space of $\mathbf{X}$).

Coefficient of Multiple Determination

The *coefficient of multiple determination* is defined as

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

and gives the amount of variation in y_i that is explained by the linear relationships with $x_{i1}, \ldots, x_{ip}$.

When interpreting R^2 values, note that...

- $0 \leq R^2 \leq 1$
- Large R^2 values do not necessarily imply a good model

Adjusted Coefficient of Multiple Determination (R_a^2)

Including more predictors in a MLR model can artificially inflate R^2:

- Capitalizing on spurious effects present in noisy data
- Phenomenon of *over-fitting* the data

The *adjusted* R^2 is a relative measure of fit:

$$\begin{aligned} R_a^2 &= 1 - \frac{SSE/df_E}{SST/df_T} \\ &= 1 - \frac{\hat{\sigma}^2}{s_Y^2} \end{aligned}$$

where $s_Y^2 = \frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}$ is the sample estimate of the variance of Y.

Note: R^2 and R_a^2 have different interpretations!

ANOVA Table and Regression F Test

We typically organize the SS information into an ANOVA table:

Source	SS	df	MS	F	p-value
SSR	$\sum_{i=1}^{n}(\hat{y}-\bar{y})^2$	p	MSR	F^*	p^*
SSE	$\sum_{i=1}^{n}(y-\hat{y})^2$	$n-p-1$	MSE		
SST	$\sum_{i=1}^{n}(y-\bar{y})^2$	$n-1$			

$MSR = \frac{SSR}{p}$, $MSE = \frac{SSE}{n-p-1}$, $F^* = \frac{MSR}{MSE} \sim F_{p,n-p-1}$,
$p^* = P(F_{p,n-p-1} > F^*)$

F^*-statistic and p^*-value are testing $H_0 : b_1 = \cdots = b_p = 0$ versus $H_1 : b_k \neq 0$ for some $k \in \{1, \ldots, p\}$

Example #1: Fitted Values and Residuals

	x_1	x_2	y	$\hat{y}$	$\hat{e}$	$\hat{y}^2$	$\hat{e}^2$	y^2
	0	1	11	8.1	2.9	65.61	8.41	121
	11	5	15	18.0	−3.0	324.00	9.00	225
	11	4	13	13.6	−0.6	184.96	0.36	169
	7	3	14	12.0	2.0	144.00	4.00	196
	4	1	0	5.3	−5.3	28.09	28.09	0
	10	4	19	14.3	4.7	204.49	22.09	361
	5	4	16	17.8	−1.8	316.84	3.24	256
	8	2	8	6.9	1.1	47.61	1.21	64
$\sum$	56	24	96	96.0	0.0	1315.60	76.40	1392

Example #1: ANOVA Table and R^2

Using the results from the previous table, note that

$$SST = \sum_{i=1}^{8}(y_i - \bar{y})^2 = \sum_{i=1}^{8} y_i^2 - 8\bar{y}^2 = 1392 - 8(12^2) = 240$$
$$SSE = \sum_{i=1}^{8}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{8} \hat{e}_i^2 = 76.40$$
$$SSR = SST - SSE = 240 - 76.4 = 163.6$$

which implies that $R^2 = SSR/SST = 163.6/240 = 0.6816667$

Source	SS	df	MS	F	p-value
SSR	163.6	2	81.80	5.3534	0.0572
SSE	76.4	5	15.28		
SST	240.0	7			

Retain $H_0 : b_1 = b_2 = 0$ at $\alpha = .05$ level.

Ordinary Least Squares: First Derivative

Note that we can write the OLS problem as

$$\begin{aligned} SSE &= \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 \\ &= (\mathbf{y} - \mathbf{X}\mathbf{b})'(\mathbf{y} - \mathbf{X}\mathbf{b}) \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\mathbf{b} + \mathbf{b}'\mathbf{X}'\mathbf{X}\mathbf{b} \end{aligned}$$

Taking the first derivative of SSE with respect to $\mathbf{b}$ produces

$$\frac{\partial SSE}{\partial \mathbf{b}'} = -2\mathbf{y}'\mathbf{X} + 2\mathbf{b}'\mathbf{X}'\mathbf{X}$$

Setting to zero and solving for $\mathbf{b}$ gives

$$\hat{\mathbf{b}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

Ordinary Least Squares: Second Derivative

Taking the second derivative of SSE with respect to $\mathbf{b}$ produces

$$\frac{\partial^2 SSE}{\partial \mathbf{b} \partial \mathbf{b}'} = 2\mathbf{X}'\mathbf{X}$$

Note that $\mathbf{X}'\mathbf{X}$ is positive definite (assuming $\mathbf{x}_1, \ldots, \mathbf{x}_p$ are linearly independent), so the second order condition is fulfilled.

Therefore SSE reaches its minimum at $\hat{\mathbf{b}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$.

Ordinary Least Squares: Positive Semi-Definite Proof

To prove that $\mathbf{X}'\mathbf{X}$ is positive semi-definite note that

$$\begin{aligned} \mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w} &= \mathbf{v}'\mathbf{v} \\ &= \sum_{i=1}^{n} v_i^2 \\ &\geq 0 \end{aligned}$$

where $\mathbf{v} = \mathbf{X}\mathbf{w}$ and $\mathbf{w} = (w_1, \ldots, w_{p+1})'$.

Note: $\mathbf{w}'\mathbf{X}'\mathbf{X}\mathbf{w} > 0$ if $\mathbf{X}$ has linearly independent columns.

Relation to ML Solution (same as SLR model)

Remember that $(\mathbf{y}|\mathbf{X}) \sim \mathrm{N}(\mathbf{Xb}, \sigma^2\mathbf{I}_n)$, which implies that $\mathbf{y}$ has pdf

$$f(\mathbf{y}|\mathbf{X}, \mathbf{b}, \sigma^2) = (2\pi)^{-n/2}(\sigma^2)^{-n/2}e^{-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{Xb})'(\mathbf{y}-\mathbf{Xb})}$$

As a result, the log-likelihood of $\mathbf{b}$ given $(\mathbf{y}, \mathbf{X}, \sigma^2)$ is

$$\ln\{L(\mathbf{b}|\mathbf{y}, \mathbf{X}, \sigma^2)\} = -\frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{Xb})'(\mathbf{y} - \mathbf{Xb}) + c$$

where c is a constant that does not depend on **b**.

Relation to ML Solution (continued)

The maximum likelihood estimate (MLE) of **b** is the estimate satisfying

$$\max_{\mathbf{b}\in\mathbb{R}^{p+1}} -\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{b})'(\mathbf{y}-\mathbf{X}\mathbf{b})$$

Now, note that...

- $\max_{\mathbf{b}\in\mathbb{R}^{p+1}} -\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\mathbf{b})'(\mathbf{y}-\mathbf{X}\mathbf{b}) = \max_{\mathbf{b}\in\mathbb{R}^{p+1}} -(\mathbf{y}-\mathbf{X}\mathbf{b})'(\mathbf{y}-\mathbf{X}\mathbf{b})$
- $\max_{\mathbf{b}\in\mathbb{R}^{p+1}} -(\mathbf{y}-\mathbf{X}\mathbf{b})'(\mathbf{y}-\mathbf{X}\mathbf{b}) = \min_{\mathbf{b}\in\mathbb{R}^{p+1}}(\mathbf{y}-\mathbf{X}\mathbf{b})'(\mathbf{y}-\mathbf{X}\mathbf{b})$

Thus, the OLS and ML estimate of **b** is the same: $\hat{\mathbf{b}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$

Estimated Error Variance (Mean Squared Error)

The estimated error variance is

$$\begin{aligned}\hat{\sigma}^2 &= SSE/(n-p-1) \\ &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2/(n-p-1) \\ &= \|(\mathbf{I}_n - \mathbf{H})\mathbf{y}\|^2/(n-p-1)\end{aligned}$$

which is an unbiased estimate of error variance σ^2.

The estimate $\hat{\sigma}^2$ is the *mean squared error (MSE)* of the model.

Proof $\hat{\sigma}^2$ is Unbiased

First note that we can write *SSE* as

$$\begin{aligned}\|(\mathbf{I}_n - \mathbf{H})\mathbf{y}\|^2 &= \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{H}\mathbf{y} + \mathbf{y}'\mathbf{H}^2\mathbf{y} \\ &= \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{H}\mathbf{y}\end{aligned}$$

Now define $\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{X}\mathbf{b}$ and note that

$$\begin{aligned}\tilde{\mathbf{y}}'\tilde{\mathbf{y}} - \tilde{\mathbf{y}}'\mathbf{H}\tilde{\mathbf{y}} &= \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\mathbf{b} + \mathbf{b}'\mathbf{X}'\mathbf{X}\mathbf{b} - \mathbf{y}'\mathbf{H}\mathbf{y} + 2\mathbf{y}'\mathbf{H}\mathbf{X}\mathbf{b} - \mathbf{b}'\mathbf{X}'\mathbf{H}\mathbf{X}\mathbf{b} \\ &= \mathbf{y}'\mathbf{y} - \mathbf{y}'\mathbf{H}\mathbf{y} \\ &= SSE\end{aligned}$$

given that $\mathbf{H}\mathbf{X} = \mathbf{X}$ (note $\mathbf{H}$ is projection matrix for column space of $\mathbf{X}$).

Now use the trace trick

$$\begin{aligned}\tilde{\mathbf{y}}'\tilde{\mathbf{y}} - \tilde{\mathbf{y}}'\mathbf{H}\tilde{\mathbf{y}} &= \mathrm{tr}(\tilde{\mathbf{y}}'\tilde{\mathbf{y}}) - \mathrm{tr}(\tilde{\mathbf{y}}'\mathbf{H}\tilde{\mathbf{y}}) \\ &= \mathrm{tr}(\tilde{\mathbf{y}}\tilde{\mathbf{y}}') - \mathrm{tr}(\mathbf{H}\tilde{\mathbf{y}}\tilde{\mathbf{y}}')\end{aligned}$$

Proof $\hat{\sigma}^2$ is Unbiased (continued)

Plugging in the previous results and taking the expectation gives

$$\begin{aligned}
\mathrm{E}(\hat{\sigma}^2) &= \frac{\mathrm{E}\left[\mathrm{tr}(\tilde{\mathbf{y}}\tilde{\mathbf{y}}')\right]}{n-p-1} - \frac{\mathrm{E}\left[\mathrm{tr}(\mathbf{H}\tilde{\mathbf{y}}\tilde{\mathbf{y}}')\right]}{n-p-1} \\
&= \frac{\mathrm{tr}(\mathrm{E}\left[\tilde{\mathbf{y}}\tilde{\mathbf{y}}'\right])}{n-p-1} - \frac{\mathrm{tr}(\mathbf{H}\mathrm{E}\left[\tilde{\mathbf{y}}\tilde{\mathbf{y}}'\right])}{n-p-1} \\
&= \frac{\mathrm{tr}(\sigma^2\mathbf{I}_n)}{n-p-1} - \frac{\mathrm{tr}(\mathbf{H}\sigma^2\mathbf{I}_n)}{n-p-1} \\
&= \frac{n\sigma^2}{n-p-1} - \frac{(p+1)\sigma^2}{n-p-1} \\
&= \sigma^2
\end{aligned}$$

which completes the proof; note that $\mathrm{tr}(\mathbf{H}) = p + 1$.

ML Estimate of σ^2: Summary

Using the same arguments from the SLR notes, we have that

$$\tilde{\sigma}^2 = \hat{\mathbf{e}}'\hat{\mathbf{e}}/n$$

is the MLE of the error variance σ^2.

Reminder: this estimate derives from maximizing $\ln\{L(\sigma^2|\mathbf{y}, \mathbf{X}, \hat{\mathbf{b}})\}$ under the assumption that $e_i \stackrel{\text{iid}}{\sim} \mathrm{N}(0, \sigma^2)$.

ML Estimate of σ^2: Bias

From our previous results using $\hat{\sigma}^2$, we have that

$$\mathrm{E}(\tilde{\sigma}^2) = \frac{n-p-1}{n}\sigma^2$$

Consequently, the *bias* of the estimator $\tilde{\sigma}^2$ is given by

$$\frac{n-p-1}{n}\sigma^2 - \sigma^2 = -\frac{p+1}{n}\sigma^2$$

and note that $-\frac{p+1}{n}\sigma^2 \to 0$ as $n \to \infty$.

Comparing $\hat{\sigma}^2$ and $\tilde{\sigma}^2$

Reminder: the MSE and MLE of σ^2 are given by

$$\hat{\sigma}^2 = \|(\mathbf{I}_n - \mathbf{H})\mathbf{y}\|^2/(n - p - 1)$$
$$\tilde{\sigma}^2 = \|(\mathbf{I}_n - \mathbf{H})\mathbf{y}\|^2/n$$

From the definitions of $\hat{\sigma}^2$ and $\tilde{\sigma}^2$ we have that

$$\tilde{\sigma}^2 < \hat{\sigma}^2$$

so the MLE produces a smaller estimate of the error variance.

Confidence Interval for σ^2

Note that $\frac{(n-p-1)\hat{\sigma}^2}{\sigma^2} = \frac{SSE}{\sigma^2} = \frac{\sum_{i=1}^{n} \hat{e}_i^2}{\sigma^2} \sim \chi^2_{n-p-1}$

This implies that

$$\chi^2_{(n-p-1;1-\alpha/2)} < \frac{(n-p-1)\hat{\sigma}^2}{\sigma^2} < \chi^2_{(n-p-1;\alpha/2)}$$

where $P(Q > \chi^2_{(n-p-1;\alpha/2)}) = \alpha/2$, so a $100(1-\alpha)$% CI is given by

$$\frac{(n-p-1)\hat{\sigma}^2}{\chi^2_{(n-p-1;\alpha/2)}} < \sigma^2 < \frac{(n-p-1)\hat{\sigma}^2}{\chi^2_{(n-p-1;1-\alpha/2)}}$$

Example #1: Calculating $\hat{\sigma}^2$ and $\tilde{\sigma}^2$

Returning to Bob's Used Cars example:

Source	SS	df	MS	F	p-value
SSR	163.6	2	81.80	5.3534	0.0572
SSE	76.4	5	15.28		
SST	240.0	7			

So the estimates of the error variance are given by

$$\hat{\sigma}^2 = MSE = 15.28$$
$$\tilde{\sigma}^2 = (5/8)MSE = 9.55$$

Summary of Results

Using the arguments from the SLR model, we have

$$\hat{\mathbf{b}} \sim \mathrm{N}(\mathbf{b}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$$

$$\hat{\mathbf{y}} \sim \mathrm{N}(\mathbf{X}\mathbf{b}, \sigma^2\mathbf{H})$$

$$\hat{\mathbf{e}} \sim \mathrm{N}(\mathbf{0}, \sigma^2(\mathbf{I}_n - \mathbf{H}))$$

Typically σ^2 is unknown, so we use the MSE $\hat{\sigma}^2$ in practice.

Inferences about $\hat{b}_j$ with σ^2 Known

If σ^2 is known, form $100(1-\alpha)\%$ CIs using

$$\hat{b}_0 \pm Z_{\alpha/2}\sigma_{b_0} \qquad \hat{b}_j \pm Z_{\alpha/2}\sigma_{b_j}$$

where

- $Z_{\alpha/2}$ is normal quantile such that $P(X > Z_{\alpha/2}) = \alpha/2$
- σ_{b_0} and σ_{b_j} are square-roots of diagonals of $\mathrm{V}(\hat{\mathbf{b}}) = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$

To test $H_0 : b_j = b_j^*$ vs. $H_1 : b_j \neq b_j^*$ (for some $j \in \{0, 1, \ldots, p\}$) use

$$Z = (\hat{b}_j - b_j^*)/\sigma_{b_j}$$

which follows a standard normal distribution under H_0.

Inferences about $\hat{b}_j$ with σ^2 Unknown

If σ^2 is unknown, form $100(1-\alpha)\%$ CIs using

$$\hat{b}_0 \pm t_{n-p-1}^{(\alpha/2)} \hat{\sigma}_{b_0} \qquad \hat{b}_j \pm t_{n-p-1}^{(\alpha/2)} \hat{\sigma}_{b_j}$$

where

- $t_{n-p-1}^{(\alpha/2)}$ is t_{n-p-1} quantile with $P(X > t_{n-p-1}^{(\alpha/2)}) = \alpha/2$
- $\hat{\sigma}_{b_0}$ and $\hat{\sigma}_{b_j}$ are square-roots of diagonals of $\hat{\mathrm{V}}(\hat{\mathbf{b}}) = \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}$

To test $H_0 : b_j = b_j^*$ vs. $H_1 : b_j \neq b_j^*$ (for some $j \in \{0, 1, \ldots, p\}$) use

$$T = (\hat{b}_j - b_j^*)/\hat{\sigma}_{b_j}$$

which follows a t_{n-p-1} distribution under H_0.

Inferences about Multiple $\hat{b}_j$

Assume that $q < p$ and want to test if a reduced model is sufficient:

$$H_0 : b_{q+1} = b_{q+2} = \cdots = b_p = b^*$$
$$H_1 : \text{at least one } b_k \neq b^*$$

Compare the SSE for full and reduced (constrained) models:
(a) Full Model: $y_i = b_0 + \sum_{j=1}^{p} b_j x_{ij} + e_i$
(b) Reduced Model: $y_i = b_0 + \sum_{j=1}^{q} b_j x_{ij} + b^* \sum_{k=q+1}^{p} x_{ik} + e_i$

Note: set $b^* = 0$ to remove $X_{q+1}, \ldots, X_p$ from model.

Inferences about Multiple $\hat{b}_j$ (continued)

Test Statistic:

$$
\begin{aligned}
F^* &= \frac{SSE_R - SSE_F}{df_R - df_F} \div \frac{SSE_F}{df_F} \\
&= \frac{SSE_R - SSE_F}{(n-q-1)-(n-p-1)} \div \frac{SSE_F}{n-p-1} \\
&\sim F_{(p-q,n-p-1)}
\end{aligned}
$$

where

- SSE_R is sum-of-squares error for reduced model
- SSE_F is sum-of-squares error for full model
- df_R is error degrees of freedom for reduced model
- df_F is error degrees of freedom for full model

Inferences about Linear Combinations of $\hat{b}_j$

Assume that $\mathbf{c} = (c_1, \ldots, c_{p+1})'$ and want to test:

$$H_0 : \mathbf{c}'\mathbf{b} = b^*$$
$$H_1 : \mathbf{c}'\mathbf{b} \neq b^*$$

Test statistic:

$$t^* = \frac{\mathbf{c}'\hat{\mathbf{b}} - b^*}{\hat{\sigma}\sqrt{\mathbf{c}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{c}}}$$
$$\sim t_{n-p-1}$$

Example #1: Inference Questions

Returning to Bob's Used Cars example, suppose we want to...

(b) Test the significance of the regression at $\alpha = .05$ and $\alpha = .1$.

(c) Test if there is a significant relationship between hours of required work (x_1) and selling price (y) given the buying price (x_2), i.e., test $H_0 : b_1 = 0$ versus $H_1 : b_1 \neq 0$. Use $\alpha = .05$ level.

(d) Test if there is a significant relationship between the buying price (x_2) and selling price (y) given the hours of required work (x_1), i.e., test $H_0 : b_2 = 0$ versus $H_1 : b_2 \neq 0$. Use $\alpha = .05$ level.

Example #1: Answer 1b

Question: Test the significance of the regression at $\alpha = .05$ and $\alpha = .1$.

The ANOVA Table for Bob's Used Cars example is:

Source	SS	df	MS	F	p-value
SSR	163.6	2	81.80	5.3534	0.0572
SSE	76.4	5	15.28		
SST	240.0	7			

The p-value is $p = 0.0572$ so we accept $H_0 : b_1 = b_2 = 0$ at $\alpha = .05$ but reject H_0 at $\alpha = .1$.

Example #1: Answer 1c

Question: Test $H_0 : b_1 = 0$ versus $H_1 : b_1 \neq 0$. Use $\alpha = .05$ level.

The covariance matrix of $\hat{\mathbf{b}}$ is given by

$$\begin{aligned}\hat{V}(\hat{\mathbf{b}}) &= \hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1} \\ &= 15.28 \begin{pmatrix} 0.7125 & -0.025 & -0.1375 \\ -0.025 & 0.025 & -0.05 \\ -0.1375 & -0.05 & 0.1625 \end{pmatrix}\end{aligned}$$

so $\hat{\sigma}_{\hat{b}_1} = \sqrt{15.28(0.025)} = 0.6180615$ is the standard error of $\hat{b}_1$

Example #1: Answer 1c (continued)

Question: Test $H_0 : b_1 = 0$ versus $H_1 : b_1 \neq 0$. Use $\alpha = .05$ level.

The t test statistic is given by $T = \frac{\hat{b}_1}{\hat{\sigma}_{\hat{b}_1}} = \frac{-0.7}{0.6180615} = -1.132573$

The critical t values are given by $t_5^{(.975)} = -2.570582$ and $t_5^{(.025)} = 2.570582$, so the decision is

$$t_5^{(.975)} = -2.570582 < -1.132573 = T \Longrightarrow \text{Retain } H_0$$

Example #1: Answer 1d

Question: Test $H_0 : b_2 = 0$ versus $H_1 : b_2 \neq 0$. Use $\alpha = .05$ level.

$\hat{\sigma}_{\hat{b}_2} = \sqrt{15.28(0.1625)} = 1.575754$ is the standard error of $\hat{b}_2$

The t test statistic is given by $T = \frac{\hat{b}_2}{\hat{\sigma}_{\hat{b}_2}} = \frac{4.4}{1.575754} = 2.792314$

The critical t values are given by $t_5^{(.975)} = -2.570582$ and $t_5^{(.025)} = 2.570582$, so the decision is

$$t_5^{(.025)} = 2.570582 < 2.792314 = T \Longrightarrow \text{Reject } H_0$$

Interval Estimation

Idea: estimate *expected value of response* for a given predictor score.

Given $\mathbf{x}_h = (1, x_{h1}, \ldots, x_{hp})$, the fitted value is $\hat{y}_h = \mathbf{x}_h\hat{\mathbf{b}}$.

Variance of $\hat{y}_h$ is given by $\sigma^2_{\bar{y}_h} = \mathrm{V}(\mathbf{x}_h\hat{\mathbf{b}}) = \mathbf{x}_h\mathrm{V}(\hat{\mathbf{b}})\mathbf{x}'_h = \sigma^2\mathbf{x}_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}'_h$

- Use $\hat{\sigma}^2_{\bar{y}_h} = \hat{\sigma}^2\mathbf{x}_h(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}'_h$ if σ^2 is unknown

We can test $H_0 : \mathrm{E}(y_h) = y_h^*$ vs. $H_1 : \mathrm{E}(y_h) \neq y_h^*$

- Test statistic: $T = (\hat{y}_h - y_h^*)/\hat{\sigma}_{\bar{y}_h}$, which follows t_{n-p-1} distribution
- $100(1-\alpha)\%$ CI for $E(y_h)$: $\hat{y}_h \pm t_{n-p-1}^{(\alpha/2)}\hat{\sigma}_{\bar{y}_h}$

Predicting New Observations

Idea: estimate *observed value of response* for a given predictor score.

- Note: interested in actual $\hat{y}_h$ value instead of $\mathrm{E}(\hat{y}_h)$

Given $\mathbf{x}_h = (1, x_{h1}, \ldots, x_{hp})$, the fitted value is $\hat{y}_h = \mathbf{x}_h\hat{\mathbf{b}}$.

- Note: same as interval estimation

When predicting a new observation, there are two uncertainties:

- location of the distribution of Y for $X_1, \ldots, X_p$ (captured by $\sigma^2_{\hat{y}_h}$)
- variability within the distribution of Y (captured by σ^2)

Predicting New Observations (continued)

Two sources of variance are independent so $\sigma^2_{y_h} = \sigma^2_{\bar{y}_h} + \sigma^2$

- Use $\hat{\sigma}^2_{y_h} = \hat{\sigma}^2_{\bar{y}_h} + \hat{\sigma}^2$ if σ^2 is unknown

We can test $H_0 : y_h = y_h^*$ vs. $H_1 : y_h \neq y_h^*$

- Test statistic: $T = (\hat{y}_h - y_h^*)/\hat{\sigma}_{y_h}$, which follows t_{n-p-1} distribution
- $100(1-\alpha)\%$ Prediction Interval (PI) for y_h: $\hat{y}_h \pm t_{n-p-1}^{(\alpha/2)} \hat{\sigma}_{y_h}$

Simultaneous Confidence Regions

In MLR we typically want a *confidence region*, which is similar to a CI but holds for multiple coefficients (i.e, b_j) simultaneously.

Given the distribution of $\hat{\mathbf{b}}$ (and some probability theory), we have that

$$\frac{(\hat{\mathbf{b}} - \mathbf{b})'\mathbf{X}'\mathbf{X}(\hat{\mathbf{b}} - \mathbf{b})}{\sigma^2} \sim \chi^2_{p+1}$$

$$\frac{(n-p-1)\hat{\sigma}^2}{\sigma^2} \sim \chi^2_{n-p-1}$$

which implies that

$$\frac{(\hat{\mathbf{b}} - \mathbf{b})'\mathbf{X}'\mathbf{X}(\hat{\mathbf{b}} - \mathbf{b})}{(p+1)\hat{\sigma}^2} \sim \frac{\chi^2_{p+1}/(p+1)}{\chi^2_{n-p-1}/(n-p-1)} \equiv F_{(p+1,n-p-1)}$$

Simultaneous Confidence Regions (continued)

To form a $100(1-\alpha)\%$ confidence region (CR) use limits such that

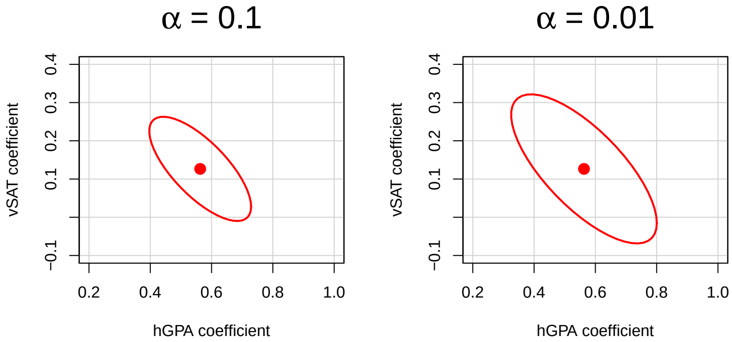
$$(\hat{\mathbf{b}} - \mathbf{b})'\mathbf{X}'\mathbf{X}(\hat{\mathbf{b}} - \mathbf{b}) \leq (p+1)\hat{\sigma}^2 F^{(\alpha)}_{(p+1, n-p-1)}$$

where $F^{(\alpha)}_{(p+1, n-p-1)}$ is the critical value for significance level α.

CRs are 2D ellipse with $p = 2$ and higher-dimensional ellipse for $p > 2$.

Simultaneous Confidence Regions (example)

Returning to the GPA example, the simultaneous CR for b_1, b_2 is:



Created using `car` package in R.

Note: we reject $H_0 : b_1 = b_2 = 0$ because point (0,0) is not within CR.

Example #1: Prediction Questions

Returning to Bob's Used Cars example, suppose we want to...

(e) Construct a 90% prediction interval for the value of Y at $x_1 = 2$ and $x_2 = 3$

(f) Construct a 90% prediction interval for the value of Y at $x_1 = 8$ and $x_2 = 5$

Example #1: Answer 1e

Question: Construct a 90% prediction interval for the value of Y at $x_1 = 2$ and $x_2 = 3$

Predicted value: $\hat{y} = 3.7 - 0.7x_1 + 4.4x_2 = 3.7 - 0.7(2) + 4.4(3) = 15.5$

The variance of a new observation with $x_1 = 2$ and $x_2 = 3$ is

$$\begin{aligned}
\hat{\sigma}^2_{\hat{y}} &= \hat{\sigma}^2 \left[1 + \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} (\mathbf{X}'\mathbf{X})^{-1} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \right] \\
&= 15.28 \left[1 + \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 0.7125 & -0.025 & -0.1375 \\ -0.025 & 0.025 & -0.05 \\ -0.1375 & -0.05 & 0.1625 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \right] \\
&= 15.28[1 + 0.75] \\
&= 26.74
\end{aligned}$$

Example #1: Answer 1e (continued)

Question: Construct a 90% prediction interval for the value of Y at $x_1 = 2$ and $x_2 = 3$

The critical t_5 values are $t_5^{(.95)} = -2.015048$ and $t_5^{(.05)} = 2.015048$

So the 90% PI is given by

$$\begin{aligned} \hat{y} \pm t_5^{(.05)} \hat{\sigma}_{\hat{y}} &= 15.5 \pm 2.015048\sqrt{26.74} \\ &= [5.080039;\ 25.91996] \end{aligned}$$

Example #1: Answer 1f

Question: Construct a 90% prediction interval for the value of Y at $x_1 = 8$ and $x_2 = 5$

Predicted value: $\hat{y} = 3.7 - 0.7x_1 + 4.4x_2 = 3.7 - 0.7(8) + 4.4(5) = 20.1$

The variance of a new observation with $x_1 = 8$ and $x_2 = 5$ is

$$
\begin{aligned}
\hat{\sigma}^2_{\hat{y}} &= \hat{\sigma}^2 \left[1 + \begin{pmatrix} 1 & 8 & 5 \end{pmatrix} (\mathbf{X}'\mathbf{X})^{-1} \begin{pmatrix} 1 \\ 8 \\ 5 \end{pmatrix} \right] \\
&= 15.28 \left[1 + \begin{pmatrix} 1 & 8 & 5 \end{pmatrix} \begin{pmatrix} 0.7125 & -0.025 & -0.1375 \\ -0.025 & 0.025 & -0.05 \\ -0.1375 & -0.05 & 0.1625 \end{pmatrix} \begin{pmatrix} 1 \\ 8 \\ 5 \end{pmatrix} \right] \\
&= 15.28[1 + 0.6] \\
&= 24.448
\end{aligned}
$$

Example #1: Answer 1f (continued)

Question: Construct a 90% prediction interval for the value of Y at $x_1 = 8$ and $x_2 = 5$

The critical t_5 values are $t_5^{(.95)} = -2.015048$ and $t_5^{(.05)} = 2.015048$

So the 90% PI is given by

$$\begin{aligned} \hat{y} \pm t_5^{(.05)} \hat{\sigma}_{\hat{y}} &= 20.1 \pm 2.015048\sqrt{24.448} \\ &= [10.13661;\ 30.06339] \end{aligned}$$

GPA Data: Source

This example uses the *GPA* data set that we examined before.

- From http://onlinestatbook.com/2/regression/intro.html

Y: student's university grade point average.

Possible predictor variables include

- X_1: student's high school grade point average
- X_2: student's verbal SAT score
- X_3: student's math SAT score

Have data from $n = 105$ different students.

GPA Data: Summary

Summary statistics for GPA data set:

```
> summary(gpa[,1:3])
    high_GPA          math_SAT          verb_SAT    
 Min.   :2.030   Min.   :516.0   Min.   :480.0  
 1st Qu.:2.670   1st Qu.:573.0   1st Qu.:548.0  
 Median :3.170   Median :612.0   Median :591.0  
 Mean   :3.076   Mean   :623.1   Mean   :598.6  
 3rd Qu.:3.480   3rd Qu.:675.0   3rd Qu.:645.0  
 Max.   :4.000   Max.   :718.0   Max.   :732.0  
```

Note that SAT scores have a very different scales (than HS GPA).

- 1-unit change in GPA is a big difference
- 1-unit change in SAT scores is a small difference

GPA Data: Rescaling

To make regression coefficients more interpretable, rescale SAT scores by dividing them by 100 points:

```
> gpa[,2:3]=gpa[,2:3]/100
> summary(gpa[,1:3])
    high_GPA         math_SAT         verb_SAT
 Min.   :2.030   Min.   :5.160   Min.   :4.800
 1st Qu.:2.670   1st Qu.:5.730   1st Qu.:5.480
 Median :3.170   Median :6.120   Median :5.910
 Mean   :3.076   Mean   :6.231   Mean   :5.986
 3rd Qu.:3.480   3rd Qu.:6.750   3rd Qu.:6.450
 Max.   :4.000   Max.   :7.180   Max.   :7.320
```

GPA Analyses: Full Model

```
> gpaFmod=lm(univ_GPA~high_GPA+verb_SAT+math_SAT,data=gpa)
> summary(gpaFmod)

Call:
lm(formula = univ_GPA ~ high_GPA + verb_SAT + math_SAT, data = gpa)

Residuals:
     Min       1Q   Median       3Q      Max
-0.68186 -0.13189  0.01289  0.16186  0.93994

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.57935    0.34226   1.693   0.0936 .
high_GPA     0.54542    0.08503   6.415  4.6e-09 ***
verb_SAT     0.10202    0.08123   1.256   0.2120
math_SAT     0.04893    0.10215   0.479   0.6330
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2784 on 101 degrees of freedom
Multiple R-squared:  0.6236, Adjusted R-squared:  0.6124
F-statistic: 55.77 on 3 and 101 DF,  p-value: < 2.2e-16
```

GPA Analyses: Reduced Model (Dropping Math SAT)

```
> gpaRmod=update(gpaFmod,~.-math_SAT)
> summary(gpaRmod)

Call:
lm(formula = univ_GPA ~ high_GPA + verb_SAT, data = gpa)

Residuals:
     Min       1Q   Median       3Q      Max
-0.68430 -0.11268  0.01802  0.14901  0.95239

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.68387    0.26267   2.604   0.0106 *
high_GPA     0.56283    0.07657   7.350 5.07e-11 ***
verb_SAT     0.12654    0.06283   2.014   0.0466 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2774 on 102 degrees of freedom
Multiple R-squared:  0.6227, Adjusted R-squared:  0.6153
F-statistic: 84.18 on 2 and 102 DF,  p-value: < 2.2e-16
```

GPA Analyses: ANOVA Table

Use the `anova` function to compare full and reduced models:

```
> anova(gpaRmod,gpaFmod)
Analysis of Variance Table

Model 1: univ_GPA ~ high_GPA + verb_SAT
Model 2: univ_GPA ~ high_GPA + verb_SAT + math_SAT
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    102 7.8466
2    101 7.8288  1  0.017783 0.2294  0.633
```

Note: no significant difference between SSE of full and reduced models at the $\alpha = .05$ level, so we'll drop math SAT predictor.

GPA Analyses: ANOVA Table (continued)

Or use the `anova` function to get sequential sum-of-squares tests:

```
> anova(gpaRmod)
Analysis of Variance Table

Response: univ_GPA
           Df  Sum Sq Mean Sq  F value   Pr(>F)
high_GPA    1 12.6394 12.6394 164.3026 < 2e-16 ***
verb_SAT    1  0.3121  0.3121   4.0571 0.04662 *
Residuals 102  7.8466  0.0769
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation: `high_GPA` is significant at $\alpha = .001$ level, and given `high_GPA` the `verb_SAT` is significant at $\alpha = .05$ (but not at $\alpha = .01$).

GPA Analyses: ANOVA Table (continued)

Note that order of effects matters with sequential SS:

```
> gpa2mod=lm(univ_GPA~verb_SAT+high_GPA,data=gpa)
> anova(gpa2mod)
Analysis of Variance Table

Response: univ_GPA
           Df Sum Sq Mean Sq F value    Pr(>F)
verb_SAT    1 8.7954  8.7954 114.333 < 2.2e-16 ***
high_GPA    1 4.1562  4.1562  54.027 5.067e-11 ***
Residuals 102 7.8466  0.0769
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation: `verb_SAT` is significant at $\alpha = .001$ level, and given `verb_SAT` the `high_GPA` is still significant at $\alpha = .001$.

GPA Analyses: Test Multiple Slopes

To test $H_0 : b_1 = b_2$ versus $H_1 : b_1 \neq b_2$, you can use:

```
> xvar=gpa$high_GPA+gpa$verb_SAT
> gpaEmod=lm(univ_GPA~xvar,data=gpa)
> anova(gpaEmod,gpaRmod)
Analysis of Variance Table

Model 1: univ_GPA ~ xvar
Model 2: univ_GPA ~ high_GPA + verb_SAT
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1    103 8.7184
2    102 7.8466  1   0.87176 11.332 0.001075 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note: significant difference between SSE of full and reduced models at the $\alpha = .05$ level, so reject H_0.

GPA Analyses: Test Multiple Slopes (continued)

To test $H_0 : b_0 = b_1$ versus $H_1 : b_0 \neq b_1$, you can use:

```
> high_GPA1p=1+gpa$high_GPA
> gpaImod=lm(univ_GPA~0+high_GPA1p+verb_SAT,data=gpa)
> gpaImod$coef
high_GPA1p   verb_SAT
 0.5680703  0.1429841
> gpaRmod$coef
(Intercept)    high_GPA    verb_SAT
  0.6838723   0.5628331   0.1265445
```

GPA Analyses: Test Multiple Slopes (continued)

Continuing with the test of $H_0 : b_0 = b_1$ versus $H_1 : b_0 \neq b_1$:

```
> anova(gpaImod,gpaRmod)
Analysis of Variance Table

Model 1: univ_GPA ~ 0 + high_GPA1p + verb_SAT
Model 2: univ_GPA ~ high_GPA + verb_SAT
  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1    103 7.8629
2    102 7.8466  1  0.016307 0.212 0.6462
```

Note: no significant difference between SSE of full and reduced models at the $\alpha = .05$ level, so retain H_0.

GPA Analyses: Linear Combinations

To test $H_0 : b_1 - 3b_2 = 0$ versus $H_1 : b_1 - 3b_2 \neq 0$, you can use:

```
> wvar=gpa$high_GPA+gpa$verb_SAT/3
> gpaLmod=lm(univ_GPA~wvar,data=gpa)
> anova(gpaLmod,gpaRmod)
Analysis of Variance Table

Model 1: univ_GPA ~ wvar
Model 2: univ_GPA ~ high_GPA + verb_SAT
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    103 7.8880
2    102 7.8466  1  0.041411 0.5383 0.4648
```

Note: no significant difference between SSE of full and reduced models at the $\alpha = .05$ level, so retain H_0.

GPA Results: Coefficients

To examine the table of coefficients and standard errors use:

```
> sumRmod=summary(gpaRmod)
> sumRmod$coef
             Estimate Std. Error  t value     Pr(>|t|)
(Intercept) 0.6838723 0.26267241 2.603518 1.060300e-02
high_GPA    0.5628331 0.07657288 7.350294 5.067057e-11
verb_SAT    0.1265445 0.06282579 2.014213 4.661979e-02
```

- $\hat{b}_0 = 0.6839$ is expected `univ_GPA` for students with `high_GPA`=0 and `verb_SAT`=0.
- $\hat{b}_1 = 0.5628$ is expected change in `univ_GPA` for student's with `high_GPA` one point higher (holding `verb_SAT` score constant)
- $\hat{b}_2 = 0.1265$ is expected change in `univ_GPA` for student's with verbal SAT 100 points higher (holding `high_GPA` constant)

GPA Results: Error Variance and R^2

To examine the estimated error variance and R^2:

```
> sumRmod$sigma
[1] 0.2773584
> sumRmod$sigma^2
[1] 0.07692768
> sumRmod$r.squared
[1] 0.6227248
> sumRmod$adj.r.squared
[1] 0.6153272
```

Estimated error variance is $\hat{\sigma}^2 = 0.0769$.

Model explains about 62% of the variation in university GPA scores.

GPA Analyses: Manual Calculations (F model)

```
> XF=cbind(1,gpa$high_GPA,gpa$verb_SAT,gpa$math_SAT)
> y=gpa$univ_GPA
> XtXF=crossprod(XF)
> XtyF=crossprod(XF,y)
> XtXiF=solve(XtXF)
> bhatF=XtXiF%*%XtyF
> yhatF=XF%*%bhatF
> ehatF=y-yhatF
> sigsqF=sum(ehatF^2)/(nrow(XF)-ncol(XF))
> bhatseF=sqrt(sigsqF*diag(XtXiF))
> tvalF=bhatF/bhatseF
> pvalF=2*(1-pt(abs(tvalF),nrow(XF)-ncol(XF)))
> RsqF=1-sum(ehatF^2)/sum((y-mean(y))^2)
> aRsqF=1-(sum(ehatF^2)/(nrow(XF)-ncol(XF)))/(sum((y-mean(y))^2)/(nrow(XF)-1))
> data.frame(bhat=bhatF,se=bhatseF,t=tvalF,p=pvalF)
        bhat         se         t            p
1 0.57934783 0.34226274 1.6926991 9.359537e-02
2 0.54542131 0.08502654 6.4147186 4.600647e-09
3 0.10202454 0.08122676 1.2560459 2.119970e-01
4 0.04892928 0.10215357 0.4789777 6.329899e-01
> cbind(RsqF,aRsqF)
          RsqF    aRsqF
[1,] 0.6235798 0.612399
```

GPA Analyses: Manual Calculations (R model)

```
> XR=cbind(1,gpa$high_GPA,gpa$verb_SAT)
> y=gpa$univ_GPA
> XtXR=crossprod(XR)
> XtyR=crossprod(XR,y)
> XtXiR=solve(XtXR)
> bhatR=XtXiR%*%XtyR
> yhatR=XR%*%bhatR
> ehatR=y-yhatR
> sigsqR=sum(ehatR^2)/(nrow(XR)-ncol(XR))
> bhatseR=sqrt(sigsqR*diag(XtXiR))
> tvalR=bhatR/bhatseR
> pvalR=2*(1-pt(abs(tvalR),nrow(XR)-ncol(XR)))
> RsqR=1-sum(ehatR^2)/sum((y-mean(y))^2)
> aRsqR=1-(sum(ehatR^2)/(nrow(XR)-ncol(XR)))/(sum((y-mean(y))^2)/(nrow(XR)-1))
> data.frame(bhat=bhatR,se=bhatseR,t=tvalR,p=pvalR)
       bhat         se        t            p
1 0.6838723 0.26267241 2.603518 1.060300e-02
2 0.5628331 0.07657288 7.350294 5.067058e-11
3 0.1265445 0.06282579 2.014213 4.661979e-02
> cbind(RsqR,aRsqR)
          RsqR     aRsqR
[1,] 0.6227248 0.6153272
```

GPA Analyses: Manual Calculations (E model)

```
> XE=cbind(1,gpa$high_GPA+gpa$verb_SAT)
> y=gpa$univ_GPA
> XtXE=crossprod(XE)
> XtyE=crossprod(XE,y)
> XtXiE=solve(XtXE)
> bhatE=XtXiE%*%XtyE
> yhatE=XE%*%bhatE
> ehatE=y-yhatE
> sigsqE=sum(ehatE^2)/(nrow(XE)-ncol(XE))
> bhatseE=sqrt(sigsqE*diag(XtXiE))
> tvalE=bhatE/bhatseE
> pvalE=2*(1-pt(abs(tvalE),nrow(XE)-ncol(XE)))
> RsqE=1-sum(ehatE^2)/sum((y-mean(y))^2)
> aRsqE=1-(sum(ehatE^2)/(nrow(XE)-ncol(XE)))/(sum((y-mean(y))^2)/(nrow(XE)-1))
> data.frame(bhat=bhatE,se=bhatseE,t=tvalE,p=pvalE)
       bhat         se         t         p
1 0.2746940 0.24425700  1.124611 0.2633681
2 0.3198015 0.02677014 11.946203 0.0000000
> cbind(RsqE,aRsqE)
          RsqE     aRsqE
[1,] 0.5808096 0.5767398
```

GPA Analyses: Manual Calculations (I model)

```
> XI=cbind(1+gpa$high_GPA,gpa$verb_SAT)
> y=gpa$univ_GPA
> XtXI=crossprod(XI)
> XtyI=crossprod(XI,y)
> XtXiI=solve(XtXI)
> bhatI=XtXiI%*%XtyI
> yhatI=XI%*%bhatI
> ehatI=y-yhatI
> sigsqI=sum(ehatI^2)/(nrow(XI)-ncol(XI))
> bhatseI=sqrt(sigsqI*diag(XtXiI))
> tvalI=bhatI/bhatseI
> pvalI=2*(1-pt(abs(tvalI),nrow(XI)-ncol(XI)))
> RsqI=1-sum(ehatI^2)/sum((y-mean(y))^2)
> aRsqI=1-(sum(ehatI^2)/(nrow(XI)-ncol(XI)))/(sum((y-mean(y))^2)/(nrow(XI)-1))
> data.frame(bhat=bhatI,se=bhatseI,t=tvalI,p=pvalI)
       bhat         se        t            p
1 0.5680703 0.07543303 7.530791 2.000777e-11
2 0.1429841 0.05149440 2.776693 6.524903e-03
> cbind(RsqI,aRsqI)
          RsqI     aRsqI
[1,] 0.6219408 0.6182703
```

Note: R^2 values are invalid because we have no intercept in model!

GPA Analyses: Manual Calculations (L model)

```
> XL=cbind(1,gpa$high_GPA+gpa$verb_SAT/3)
> y=gpa$univ_GPA
> XtXL=crossprod(XL)
> XtyL=crossprod(XL,y)
> XtXiL=solve(XtXL)
> bhatL=XtXiL%*%XtyL
> yhatL=XL%*%bhatL
> ehatL=y-yhatL
> sigsqL=sum(ehatL^2)/(nrow(XL)-ncol(XL))
> bhatseL=sqrt(sigsqL*diag(XtXiL))
> tvalL=bhatL/bhatseL
> pvalL=2*(1-pt(abs(tvalL),nrow(XL)-ncol(XL)))
> RsqL=1-sum(ehatL^2)/sum((y-mean(y))^2)
> aRsqL=1-(sum(ehatL^2)/(nrow(XL)-ncol(XL)))/(sum((y-mean(y))^2)/(nrow(XL)-1))
> data.frame(bhat=bhatL,se=bhatseE,t=tvalL,p=pvalL)
       bhat         se         t           p
1 0.5618874 0.24425700  2.769269 0.006664712
2 0.5148101 0.02677014 12.983720 0.000000000
> cbind(RsqL,aRsqL)
          RsqL     aRsqL
[1,] 0.6207337 0.6170515
```