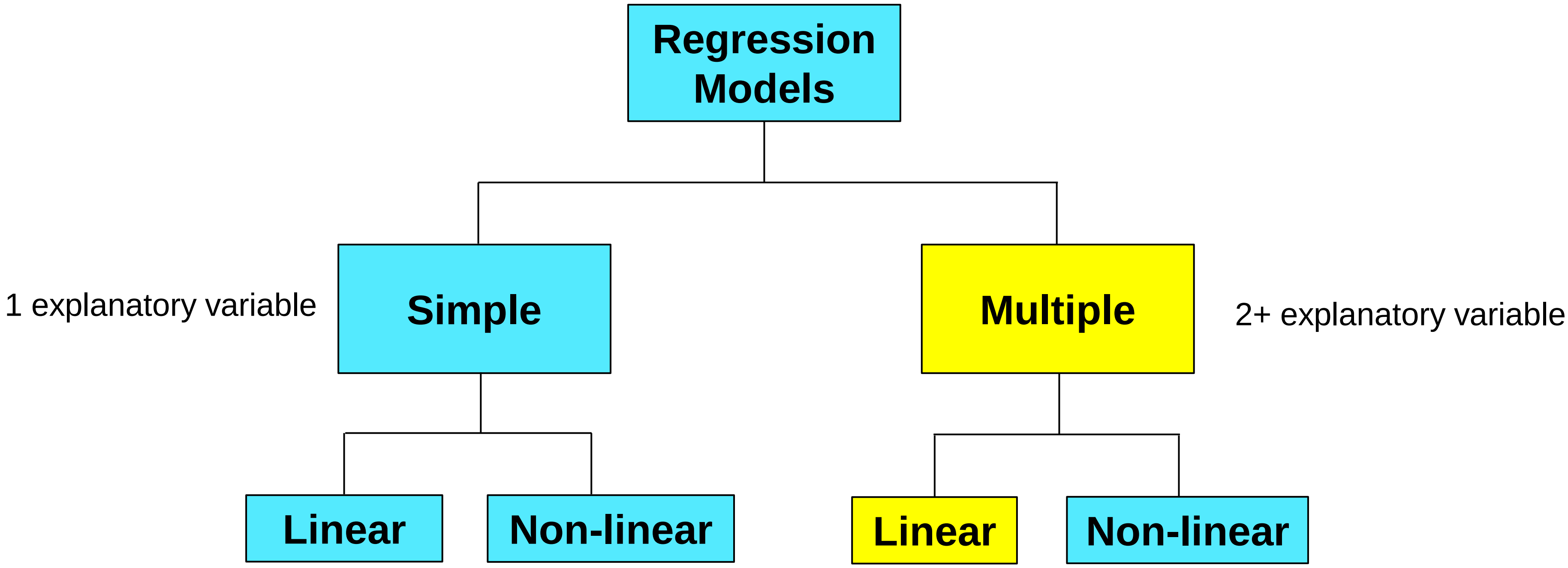
Multiple regression and correlation

Types of Regression Models



Regression modeling steps

1. **Hypothesize deterministic component**
2. Estimate unknown model parameters
3. Specify probability distribution of random error term

   estimate standard deviation of error
4. Evaluate model

Linear multiple regression model

Relationship between 1 dependent & 2 or more independent variables is a linear function

Population Y-intercept

Population slopes

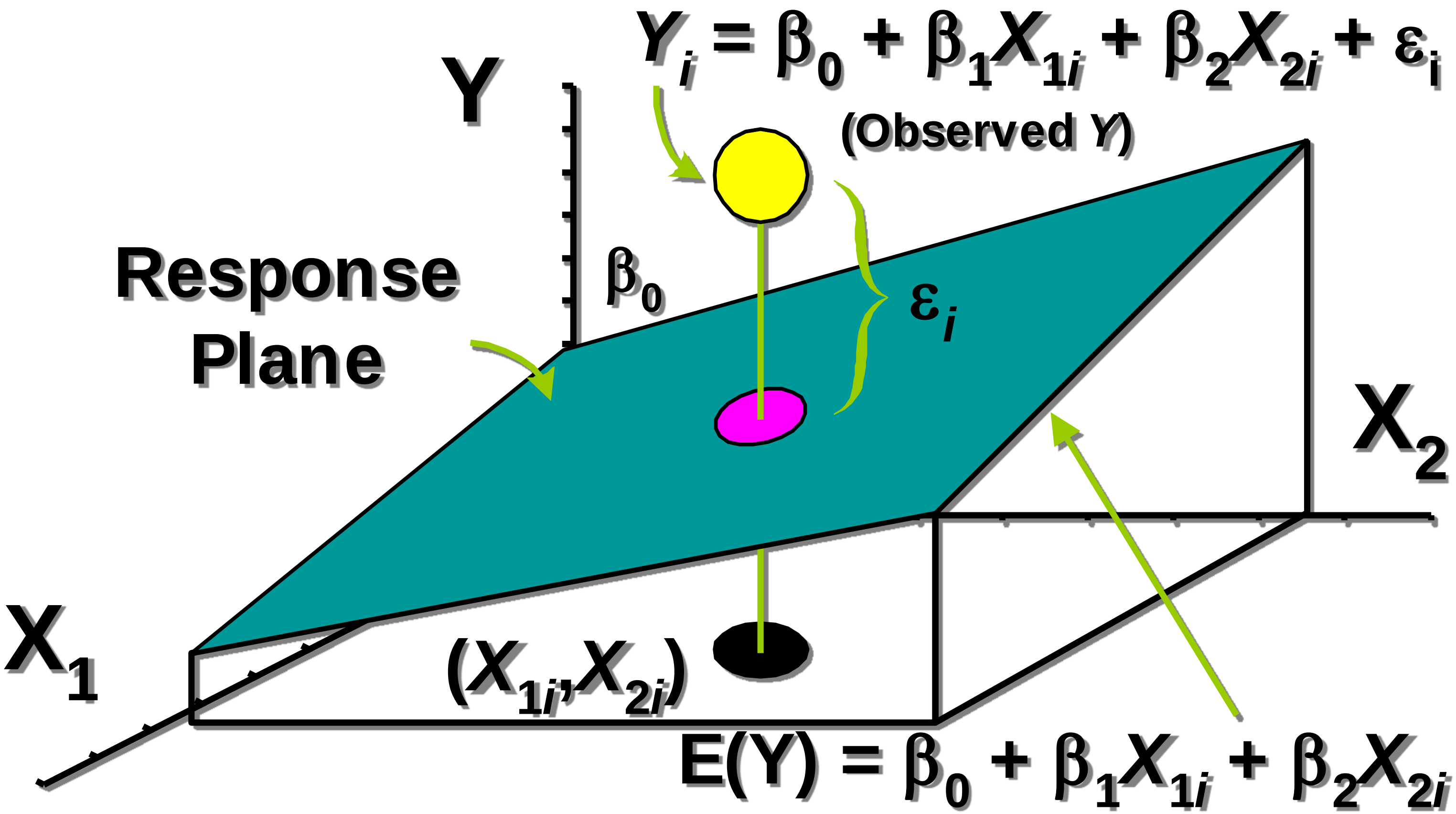
Random error

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_k X_{ki} + \varepsilon_i$$

Dependent (response) variable

Independent (explanatory) variables

Bivariate regression model



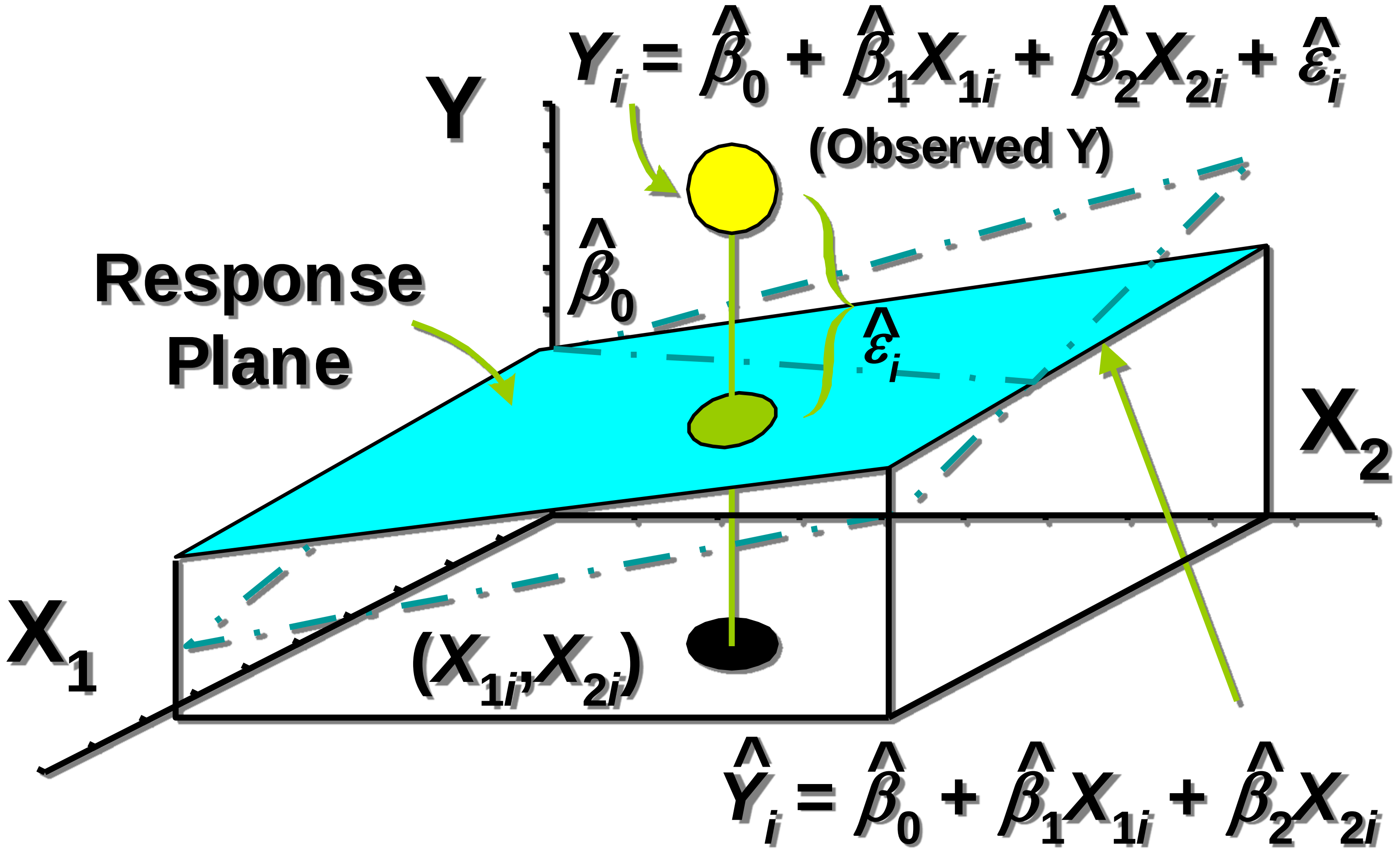
Regression modeling steps

1. Hypothesize deterministic component
2. **Estimate unknown model parameters**
3. Specify probability distribution of random error term

   Estimate standard deviation of error

4. Evaluate model

Estimate bivariate regression model



Interpretation of estimated coefficients

1. Slope ($\hat{\beta}_k$)
   – Estimated Y changes by $\hat{\beta}_k$ for each 1 unit increase in x_k ***holding all other variables constant***
     - Example: If $\hat{\beta}_1$ = 2, then sales (Y) is expected to increase by 2 for each 1 unit increase in advertising (X_1) given the number of sales rep's (X_2)

2. Y-Intercept ($\hat{\beta}_0$)
   – Average value of Y when X_k = 0

Multiple regression in matrix

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{3i} + \varepsilon_i$$

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{pmatrix} = \beta_0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} + \beta_1 \begin{pmatrix} x_{11} \\ x_{12} \\ x_{13} \\ \vdots \\ x_{1n} \end{pmatrix} + \beta_2 \begin{pmatrix} x_{21} \\ x_{22} \\ x_{23} \\ \vdots \\ x_{2n} \end{pmatrix} + \beta_3 \begin{pmatrix} x_{31} \\ x_{32} \\ x_{33} \\ \vdots \\ x_{3n} \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

$$= \begin{pmatrix} 1 & x_{11} & x_{21} & x_{31} \\ 1 & x_{12} & x_{22} & x_{32} \\ 1 & x_{13} & x_{23} & x_{33} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & x_{3n} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{pmatrix}$$

Least squares estimate (LSE)

The general multiple regression model is :

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

$$\mathbf{X} = (X_1, X_2, \cdots X_p)$$

$$X_i = (x_{1i}, x_{2i}, \cdots x_{ni})' \qquad (i = 1 \text{ to } p)$$

The LSE solution for $\boldsymbol{\beta}$ will be :

$$Min\ SSE = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_p)^2$$

In matrix notation :

$$\mathbf{X}'\mathbf{y} = \mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} \Rightarrow \hat{\boldsymbol{\beta}} = \underset{p\times p}{(\mathbf{X}'\mathbf{X})}^{-1} \underset{p\times 1}{(\mathbf{X}'\mathbf{y})}$$

$$\hat{\beta}_1 = \frac{SS_{xy}}{SS_{xx}}$$

$$X'\mathbf{y} = \begin{pmatrix} 1'\mathbf{y} \\ X_1'\mathbf{y} \\ X_2'\mathbf{y} \\ X_3'\mathbf{y} \end{pmatrix} \qquad X'X = \mathrm{SSCP} = \begin{pmatrix} 1'1 & 1'X_1 & \cdots & 1'\mathrm{X_p} \\ X_1'1 & X_1'X_1 & \cdots & X_1'X_p \\ X_2'1 & X_2'X_1 & \cdots & X_2'X_p \\ \vdots & \vdots & \ddots & \vdots \\ X_p'1 & X_p'X_1 & \cdots & X_p'X_p \end{pmatrix}$$

X' (X-prime or X-transpose)

Sum of squares and cross-products matrix (SSCP)

$$\mathbf{X} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{bmatrix}$$

$$X'X = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ b_1 & b_2 & b_3 & b_4 \\ c_1 & c_2 & c3 & c_4 \end{bmatrix} \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_4 & c_4 \end{bmatrix}$$

$$SSCP = X'X = \begin{bmatrix} \sum a_i^2 & \sum a_i b_i & \sum a_i c_i \\ \sum b_i a_i & \sum b_i^2 & \sum b_i c_i \\ \sum c_i a_i & \sum c_i b_i & \sum c_i^2 \end{bmatrix}$$

Correlation matrix and variance-covariance matrix

1	2	0
2	3	2
2	4	2
3	3	2
2	4	0
2	2	0

```
A <- matrix(c(1,2,2,3,2,2,2,3,4,3,4,2,0,2,2,2,0,0),6,3); A
SSCP <- t(A) %*% A; SSCP
```

26	37	14
37	58	20
14	20	12

```
cor(A) # correlation matrix
```

1.00	0.35	0.58
0.35	1.00	0.41
0.58	0.41	1.00

-1	-1	-1
0	0	1
0	1	1
1	0	1
0	1	-1
0	-1	-1

```
A.dev = A - rep(apply(A, 2, mean), each = length(A[,1])) # deviance
t(A.dev) %*% A.dev / (length(A[,1])-1) # variance-covariance matrix
var(A) # variance-covariance matrix
```

0.4	0.2	0.4
0.2	0.8	0.4
0.4	0.4	1.2

```
library(MASS)
ginv(SSCP) # inverse matrix
ginv(ginv(SSCP)); SSCP
ginv(A) %*% A
```

1	0	0
0	1	0
0	0	1

Fitted value and residual

The fitted value of $\mathbf{y}$, denoted $\hat{\mathbf{y}}$, is:

$$\underset{n\times 1}{\hat{\mathbf{y}}} = \mathbf{X}\hat{\boldsymbol{\beta}}$$

and the residual terms:

$$\underset{\mathbf{n\times 1}}{\mathbf{e}} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$$

we estimate residual σ^2 from sample:

$$\mathbf{s}^2(e) = MSE$$

Confidence intervals and tests of hypotheses for β

One−tailed test

$H_0: \beta_i = 0$

$H_a: \beta_i > 0$ **or** $(\beta_i < 0)$

$$s^2_{\beta_i} = \frac{\frac{1}{n-2}\sum(\mathrm{y_j} - \hat{y})^2}{\sum(x_{ij} - \bar{x}_i)^2}$$ (for simple linear regression)

test statistic: $\mathrm{t} = \frac{\beta_i}{\mathrm{s}_{\beta_i}}$

$$s^2_{\beta_i} = \frac{\frac{1}{n-k-1}\sum(\mathrm{y_j} - \hat{y})^2}{\mathrm{X'X}}$$ (for multiple linear regression)

Rejection region:

$t > t_\alpha$ (**or** $t < -t_\alpha$)

$|t| > t_{\alpha/2}$ **for two-tailed test**

$t_{\alpha/2}$ **is based on** $[\mathrm{n}-(\mathrm{p}+1)]$**df, p is number of independent variables in the model.**

Page 425 (Zar, 1999)

Parameter estimation example

The abundance (Abund) of Tibetan wild ass is associated with habitat features such as grass coverage (Cover) and elevation (Elev). We want to find the effect of these two variables.

Data

	Abund	Cover	Elev
[1,]	41	80	4835
[2,]	22	48	3216
[3,]	31	40	5012
[4,]	9	24	2818
[5,]	39	64	5201
[6,]	11	8	3678

Parameter estimation

```
Abund = c(41, 22, 31, 9, 39, 11)
Cover = c(80, 48, 40, 24, 64, 8)
Elev = c(4835, 3216, 5012, 2818, 5201, 3678)

fit = lm(Abund ~ Cover + Elev)
summary(fit)
```

$\hat{\beta}_P$

```
Coefficients:
             Estimate   Std. Error  t value Pr(>|t|)
(Intercept) -1.685e+01  2.395e+00   -7.035  0.00590 **
Cover        3.144e-01  2.715e-02   11.581  0.00138 **
Elev         6.911e-03  6.977e-04    9.905  0.00219 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 
0.1 ' ' 1
```

$\hat{\beta}_0$ $\hat{\beta}_1$ $\hat{\beta}_2$

Interpretation of coefficients solution

1. Slope ($\hat{\beta}_1$)
   - Responses to Cover is expected to increase by 0.31 individual for each 1 percent of increase in grass coverage **holding elevation constant**

2. Slope ($\hat{\beta}_2$)
   - Responses to Elev is expected to increase by 0.0069 individual for each 1 meter increase in elevation **holding coverage constant**

Regression modeling steps

1. Hypothesize deterministic component
2. Estimate unknown model parameters
3. **Specify probability distribution of random error term**

   Estimate standard deviation of error
4. Evaluate model

Variance of error

Best (unbiased) estimator of $\sigma^2 = Var(\varepsilon)$

is

$$S^2 = \frac{SSE}{n-(k+1)} = \frac{\sum \hat{\varepsilon}_i^2}{n-(k+1)}$$

Variance of error is used in formula for computing parameter variance.

$$s_{\beta_i}^2 = \frac{\frac{1}{n-k-1}\sum(\mathrm{y}_{\mathbf{j}} - \hat{y})^2}{\mathrm{X'X}}$$

where n is the number of observations, k is the number of predictors, X is the design matrix, and X'X is the transpose of X multiplied by X.

Calculating parameter variance

```
model <- lm(Volume ~ Girth + Height, data=trees)
X <- model.matrix(model); Y = trees$Volume
# Calculate the inverse of X'X
invXX = solve(t(X) %*% X) # invXX = ginv(t(X) %*% X) # library(MASS)

# Calculate the regression coefficients
beta <- invXX %*% t(X) %*% Y; beta
summary(model)

# Calculate the residuals
residuals <- Y - X %*% beta; residuals
as.data.frame(residuals(model))

# Calculate the residual variance
residual_variance <- sum(residuals^2) / (length(Y) - ncol(X))

# Calculate the standard error of the coefficients
se_beta <- sqrt(diag(residual_variance * invXX))
summary(model)
```

X'X

	(Intercept)	Girth	Height
(Intercept)	31	410.7	2356
Girth	410.7	5736.55	31524.7
Height	2356	31524.7	180274

$(X'X)^{-1}$

	(Intercept)	Girth	Height
(Intercept)	4.9519	0.0287	-0.0697
Girth	0.0287	0.0046	-0.0012
Height	-0.0697	-0.0012	0.0011

X

(Intercept)	Girth	Height
1	8.3	70
1	8.6	65
1	8.8	63
1	10.5	72
1	10.7	81
1	10.8	83
1	11	66
1	11	75
1	11.1	80
1	11.2	75
1	11.3	79
1	11.4	76
1	11.4	76
1	11.7	69
1	12	75
1	12.9	74
1	12.9	85
1	13.3	86
1	13.7	71
1	13.8	64
1	14	78
1	14.2	80
1	14.5	74
1	16	72
1	16.3	77
1	17.3	81
1	17.5	82
1	17.9	80
1	18	80
1	18	80
1	20.6	87

Regression modeling steps

1. Hypothesize deterministic component
2. Estimate unknown model parameters
3. Specify probability distribution of random error term

   Estimate standard deviation of error

4. **Evaluate model**

Evaluating multiple regression model steps

1. **Examine variation measures**
2. Do residual analysis
3. Test parameter significance

   Overall model

   Individual coefficients

4. Test for multicollinearity

Basic assumptions

- Mean value of the outcome variable for a set of explanatory variables is described by the regression equation.
- Normal distribution of values around the regression line.
- Variance around the regression line is the same for all values of the explanatory variables.
- The explanatory variables are not correlated.

Multiple coefficient of determination

- The R^2 statistic measures the overall contribution of *X*s.

$$R^2 = \frac{\text{Explained variation}}{\text{Total variation}} = \frac{SS_y - SSE}{SS_y} = 1 - \frac{SSE}{SS_y}$$

Adjusted R^2

- R^2 never decreases when new variable is added to model
  - disadvantage when comparing models

- Solution: Adjusted R^2
  - Each additional variable reduces adjusted R^2

$$R_a^2 = 1 - \left[\frac{n-1}{n-(k+1)}\right]\frac{SSE}{SS_y} \le 1 - \frac{SSE}{SS_y} = R^2$$

Evaluating multiple regression model steps

1. Examine variation measures
2. **Do residual analysis**
3. Test parameter significance
   - Overall model
   - Individual coefficients
4. Test for multicollinearity

Residual analysis

1. Graphical analysis of residuals
   - Plot estimated errors vs. X_i values
   - Plot histogram or scatter of residuals

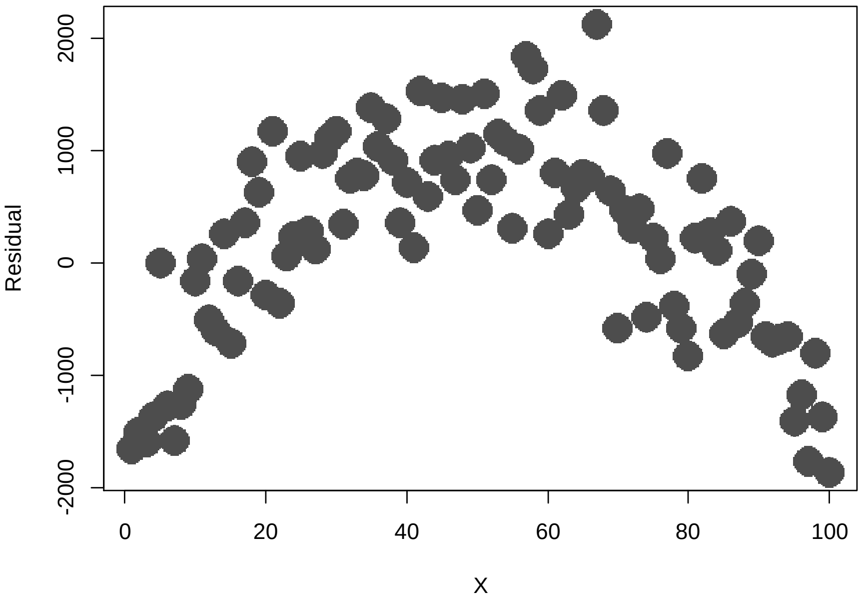
2. Purposes
   - Examine functional form (linear vs. non-linear model)
   - Evaluate violations of assumptions

Assumptions for residuals/errors

1. Mean of probability distribution of error is 0
2. Probability distribution of error has constant variance
3. Probability distribution of error is normal
4. Errors are independent

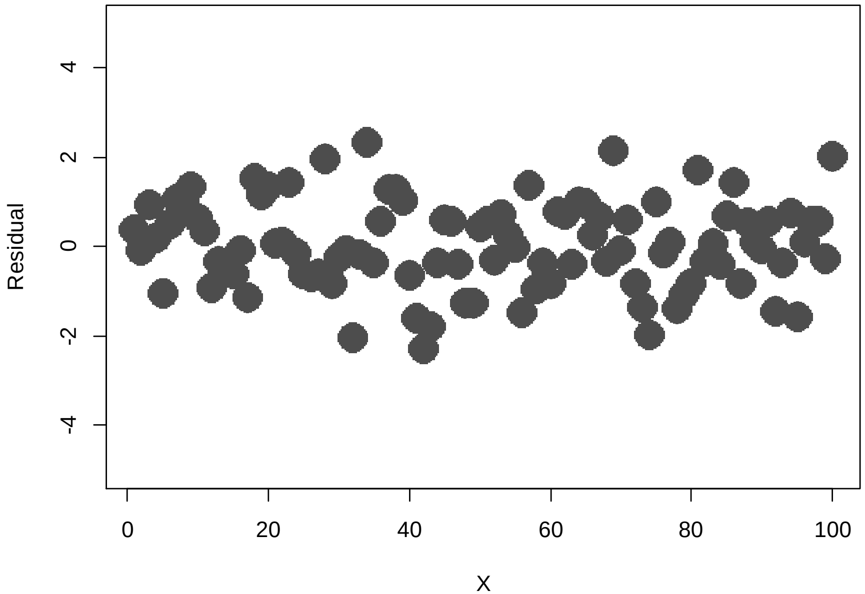
Residual plot for functional form

Add X^2 Term



X = 1:100;
Y = -(X-50)^2 + rnorm(100, 1000, 500)
plot(X, Y, cex=3, xlab='X', ylab='Residual', pch=16, col='gray30')

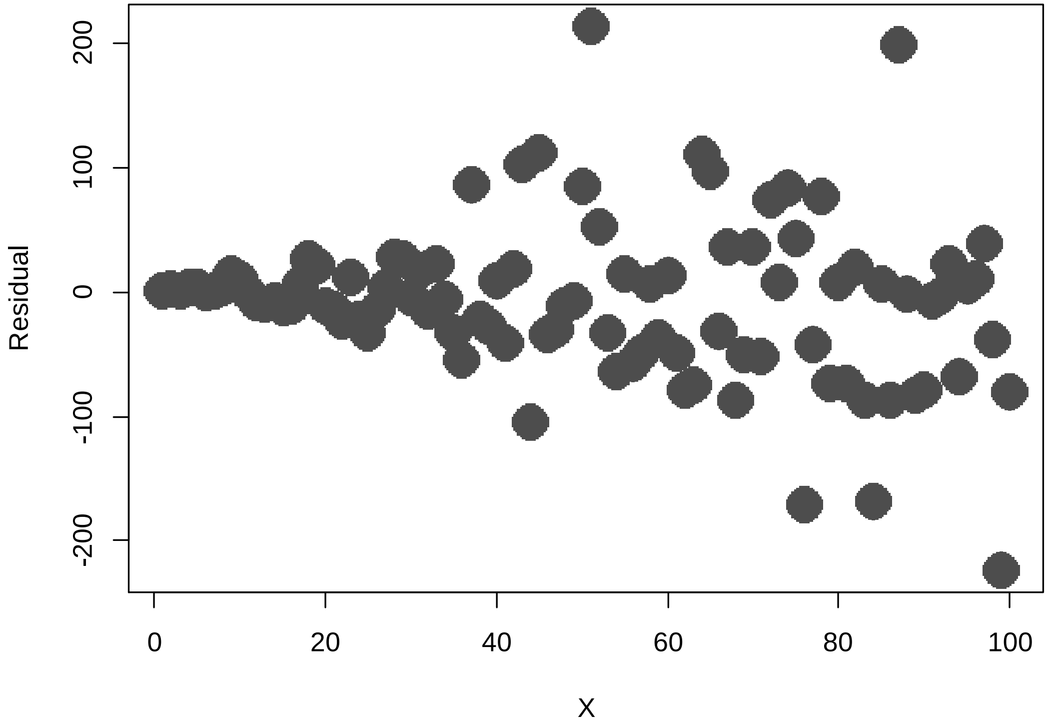
Correct Specification



X = 1:100; Y = rnorm(100, 0, 1)
plot(X, Y, ylim=c(-5,5), cex=3, xlab='X', ylab='Residual', pch=16, col='gray30')

Residual plot for equal variance

Unequal Variance

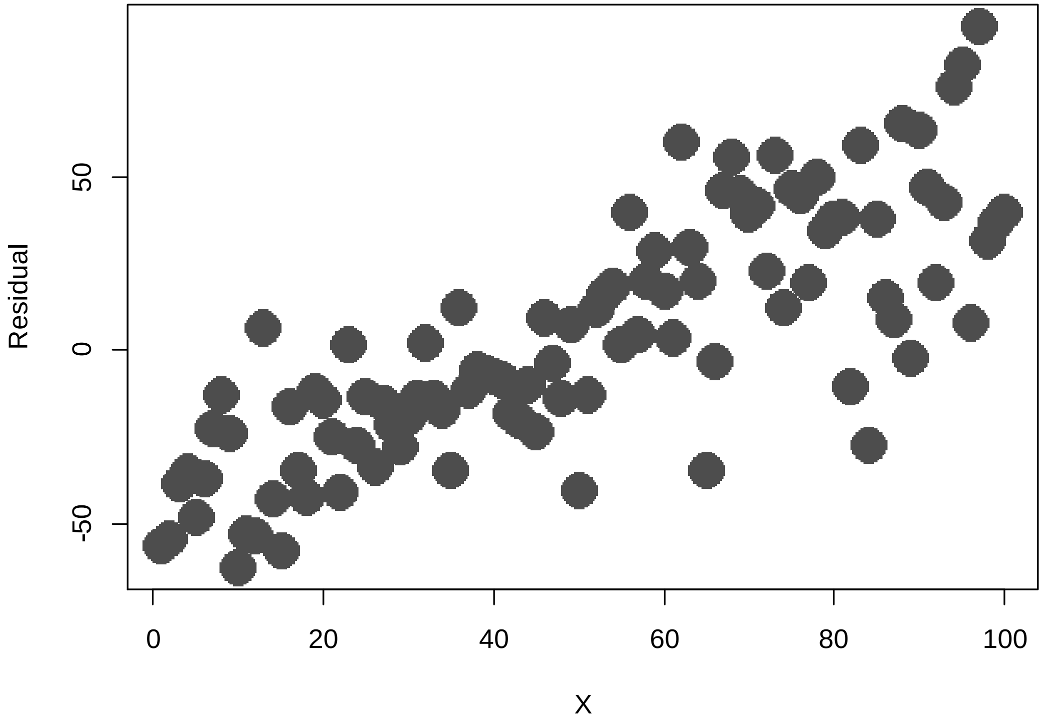


Fan-shaped

X = 1:100
Y = X*rnorm(100, 0, 1)

Residual plot for independence

Not Independent



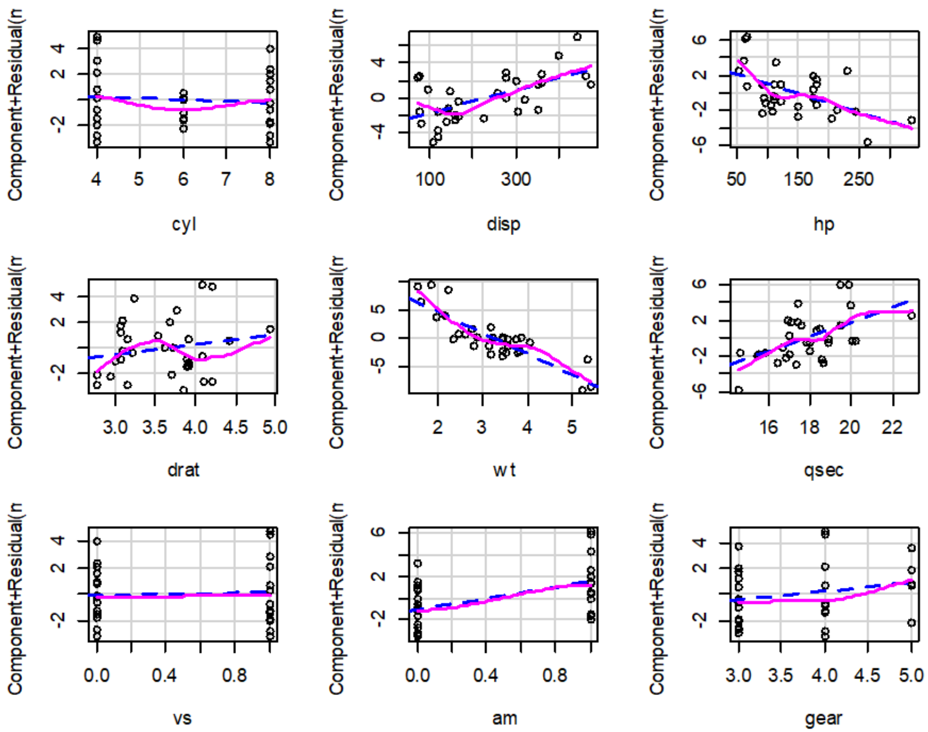
X = 1:100
Y = X+rnorm(100, 0, 20) - 50

Checking independence and linearity

```
library(car)
fit = lm(mpg ~ ., data=mtcars)
durbinWatsonTest(fit) #Durbin-Watson Test for Autocorrelated Errors
```

```
lag  Autocorrelation  D-W Statistic  p-value
  1       0.03101277       1.860893    0.342
Alternative hypothesis: rho != 0
```

```
crPlots(fit) #Component+Residual (Partial Residual) Plots
```



Evaluating multiple regression model steps

1. Examine variation measures
2. Do residual analysis
3. **Test parameter significance**
   - **Overall model**
   - Individual coefficients
4. Test for multicollinearity

Testing overall significance

1. Shows if there is a linear relationship between **all** *X* variables **together** & *Y*
2. Uses F test statistic (SSR vs. SSE)
3. Hypotheses
   - H_0: $\beta_1 = \beta_2 = ... = \beta_k = 0$
     - No Linear Relationship
   - H_a: At least one coefficient is not 0
     - At least one *X* variable affects *Y*

F Statistic for model significance

$$F = \frac{MSR}{MSE} = \frac{SSR/k}{SSE/(n-k-1)}$$

$$F = \frac{R^2/k}{(1-R^2)/(n-k-1)}$$

Rejection region: $F_{v1,v2} > F_a$, where $v_1 = k$, $v_2 = n-(k+1)$

Now the collective contribution of *X*s can be evaluated.

Confidence intervals and tests of hypotheses for β

One−tailed test

$H_0: \beta_i = 0$

$H_a: \beta_i > 0$ **or** $(\beta_i < 0)$

$$s^2_{\beta_i} = \frac{\frac{1}{n-2}\sum(\mathrm{y_j} - \hat{y})^2}{\sum(x_{ij} - \bar{x}_i)^2}$$ (for simple linear regression)

test statistic: $\mathrm{t} = \frac{\beta_i}{\mathrm{s}_{\beta_i}}$

$$s^2_{\beta_i} = \frac{\frac{1}{n-k-1}\sum(\mathrm{y_j} - \hat{y})^2}{\mathrm{X'X}}$$ (for multiple linear regression)

Rejection region:

$t > t_\alpha$ (**or** $t < -t_\alpha$)

$|t| > t_{\alpha/2}$ **for two-tailed test**

$t_{\alpha/2}$ **is based on** $[\mathrm{n} - (\mathrm{p} + 1)]$**df, p is number of independent variables in the model.**

Page 425 (Zar, 1999)

Calculating parameter variance

```
model <- lm(Volume ~ Girth + Height, data=trees)
X <- model.matrix(model); Y = trees$Volume
# Calculate the inverse of X'X
invXX = solve(t(X) %*% X) # invXX = ginv(t(X) %*% X) # library(MASS)

# Calculate the regression coefficients
beta <- invXX %*% t(X) %*% Y; beta
summary(model)

# Calculate the residuals
residuals <- Y - X %*% beta; residuals
as.data.frame(residuals(model))

# Calculate the residual variance
residual_variance <- sum(residuals^2) / (length(Y) - ncol(X))

# Calculate the standard error of the coefficients
se_beta <- sqrt(diag(residual_variance * invXX))
summary(model)
```

X'X

	(Intercept)	Girth	Height
(Intercept)	31	410.7	2356
Girth	410.7	5736.55	31524.7
Height	2356	31524.7	180274

$(X'X)^{-1}$

	(Intercept)	Girth	Height
(Intercept)	4.9519	0.0287	-0.0697
Girth	0.0287	0.0046	-0.0012
Height	-0.0697	-0.0012	0.0011

X

(Intercept)	Girth	Height
1	8.3	70
1	8.6	65
1	8.8	63
1	10.5	72
1	10.7	81
1	10.8	83
1	11	66
1	11	75
1	11.1	80
1	11.2	75
1	11.3	79
1	11.4	76
1	11.4	76
1	11.7	69
1	12	75
1	12.9	74
1	12.9	85
1	13.3	86
1	13.7	71
1	13.8	64
1	14	78
1	14.2	80
1	14.5	74
1	16	72
1	16.3	77
1	17.3	81
1	17.5	82
1	17.9	80
1	18	80
1	18	80
1	20.6	87

Model and parameter significance

```
model = lm(log(trees$Volume)~log(trees$Girth)+log(trees$Height))
summary(model)
```

```
Coefficients:
                  Estimate   Std. Error   t value   Pr(>|t|)
(Intercept)       -6.63162   0.79979      -8.292    5.06e-09 ***
log(trees$Girth)  1.98265    0.07501      26.432    < 2e-16 ***
log(trees$Height) 1.11712    0.20444      5.464     7.81e-06 ***

Residual standard error: 0.08139 on 28 degrees of freedom Multiple
R-squared: 0.9777, Adjusted R-squared: 0.9761 F-statistic: 613.2 on
2 and 28 DF, p-value: < 2.2e-16
```

Evaluating multiple regression model steps

1. Examine variation measures
2. Do residual analysis
3. Test parameter significance
   - Overall model
   - Individual coefficients
4. **Test for multicollinearity**

Multicollinearity

- High correlation between X variables
- Leads to unstable coefficients depending on X variables in model
- Always exists -- matter of degree
- Example: using both age & height as explanatory variables for weight

Two basic kinds of multicollinearity

1. **Structural multicollinearity**: This type occurs when we create a model term using other terms. In other words, it's a byproduct of the model that we specify rather than being present in the data itself. For example, if you square term X to model curvature, clearly there is a correlation between X and X^2.
2. **Data multicollinearity**: This type of multicollinearity is present in the data itself rather than being an artifact of our model. Observational experiments are more likely to exhibit this kind of multicollinearity.

Jim Frost. 2019. Regression Analysis: An Intuitive Guide [ebook]

The need to reduce multicollinearity

The need to reduce multicollinearity depends on its severity and your primary goal for your regression model.

1. **The severity of the problems increases with the degree of the multicollinearity.** Therefore, if you have only moderate multicollinearity, you may not need to resolve it.

2. **Multicollinearity affects only the specific independent variables that are correlated.** Therefore, if multicollinearity is not present for the independent variables that you are particularly interested in, you may not need to resolve it. Suppose your model contains the experimental variables of interest and some control variables. If high multicollinearity exists for the control variables but not the experimental variables, then you can interpret the experimental variables without problems.

3. **Multicollinearity affects the coefficients and p-values, but it does not influence the predictions, precision of the predictions, and the goodness-of-fit statistics.** If your primary goal is to make predictions, and you don't need to understand the role of each independent variable, you don't need to reduce severe multicollinearity.

Jim Frost. 2019. Regression Analysis: An Intuitive Guide [ebook]

Detecting multicollinearity

Examine correlation matrix

– correlations between pairs of *X* variables are more than with *Y* variable

Examine variance inflation factor (VIF)

$$VIF_j = \frac{1}{1 - R_j^2}$$

R_j^2 is the multiple correlation coefficient, the coefficient of determination of:

$$X_j = \beta_0 + \beta_1 X_1 + \ldots + \beta_{j-1} X_{j-1} + \beta_{j+1} X_{j+1} + \ldots + \beta_k X_k + \varepsilon$$

If $VIF_j > 5$ (or 10 according to text), multicollinearity exists.

Interpretation

The square root of the variance inflation factor tells you how much larger the standard error is, compared with what it would be if that variable were uncorrelated with the other independent variables in the equation.

Example

If the variance inflation factor of an independent variable were 5.27 ($\sqrt{5.27} = 2.3$) this means that the standard error for the coefficient of that independent variable is 2.3 times as large as it would be if that independent variable were uncorrelated with the other independent variables.

http://en.wikipedia.org/wiki/Variance_inflation_factor

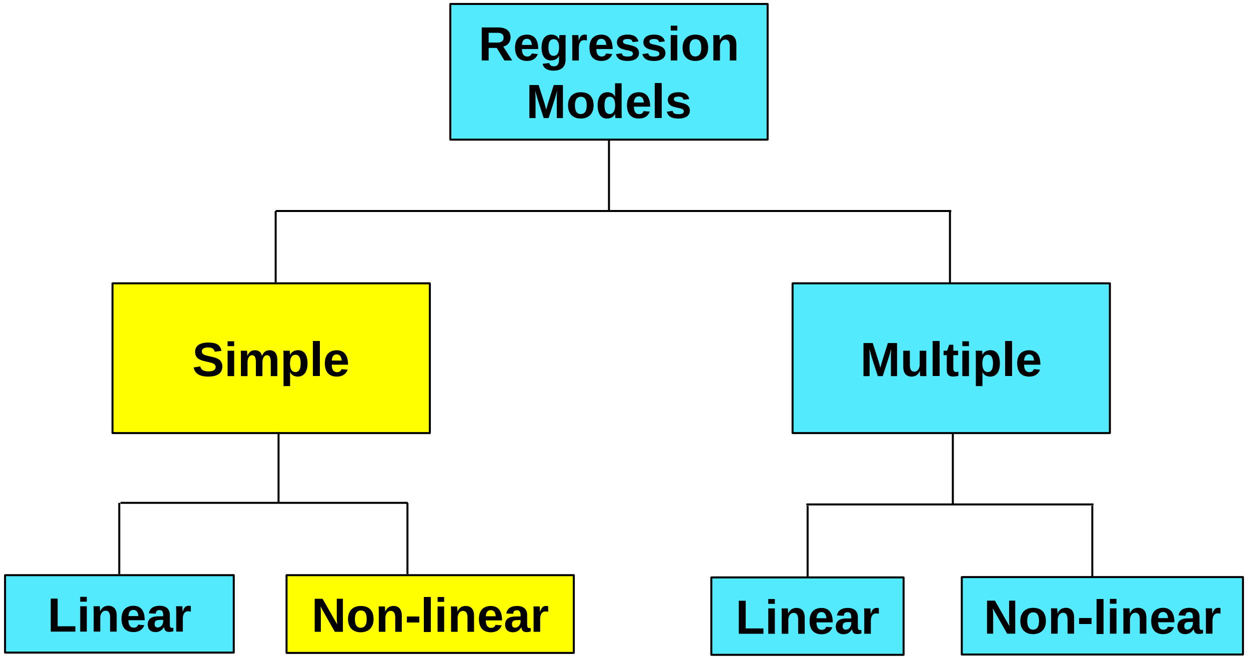
R code – VIF (variance inflation factor)

```
library(car)
vif(lm(mpg ~ ., data = mtcars))
```

cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
15.37	21.62	9.83	3.37	15.16	7.53	4.97	4.65	5.36	7.91

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1

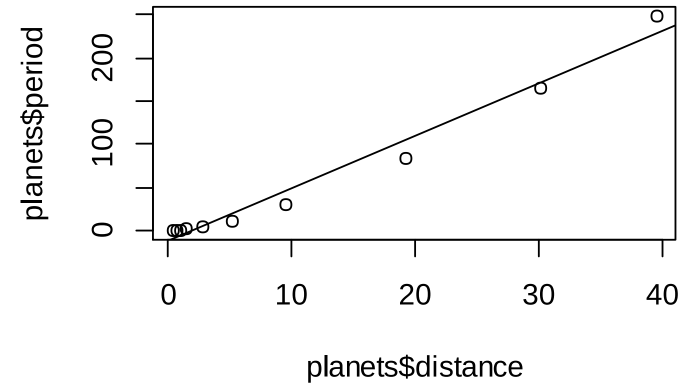
Types of Regression Models

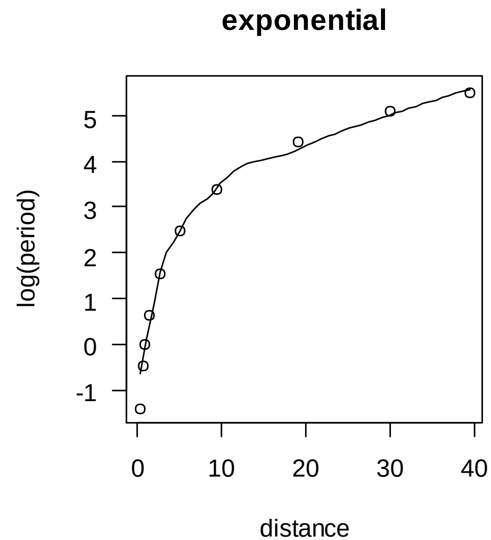


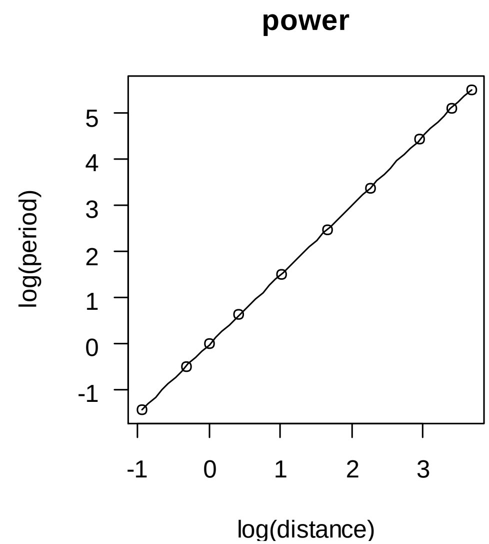
```
# Johannes Kepler's third law of planetary motion
planets = read.table(header = T, row.name = 1, text = "
planet  distance  period
Mercury   57.9      87.98
Venus    108.2     224.70
Earth     149.6     365.26
Mars     228.0     686.98
Ceres     413.8    1680.50
Jupiter    778.3    4332.00
Saturn    1427.0  10761.00
Uranus   2869.0  30685.00
Neptune 4498.0  60191.00
Pluto      5900.0  90742.00")
# units: million km, earth day

# standarized by earth
planets$distance = planets$dist / 149.6
planets$period = planets$period / 365.26

plot(planets$distance, planets$period)
abline(lm(planets$period~planets$distance))

par(mfrow=c(1,2))
with(planets, scatter.smooth(log(period) ~ distance, las=1))
title(main="exponential")
with(planets, scatter.smooth(log(period) ~ log(distance), las=1))
title(main="power")

summary(lm(log(period) ~ log(distance), data=planets))
```

Power function

$$\text{peroid}^2 = \text{distance}^3$$



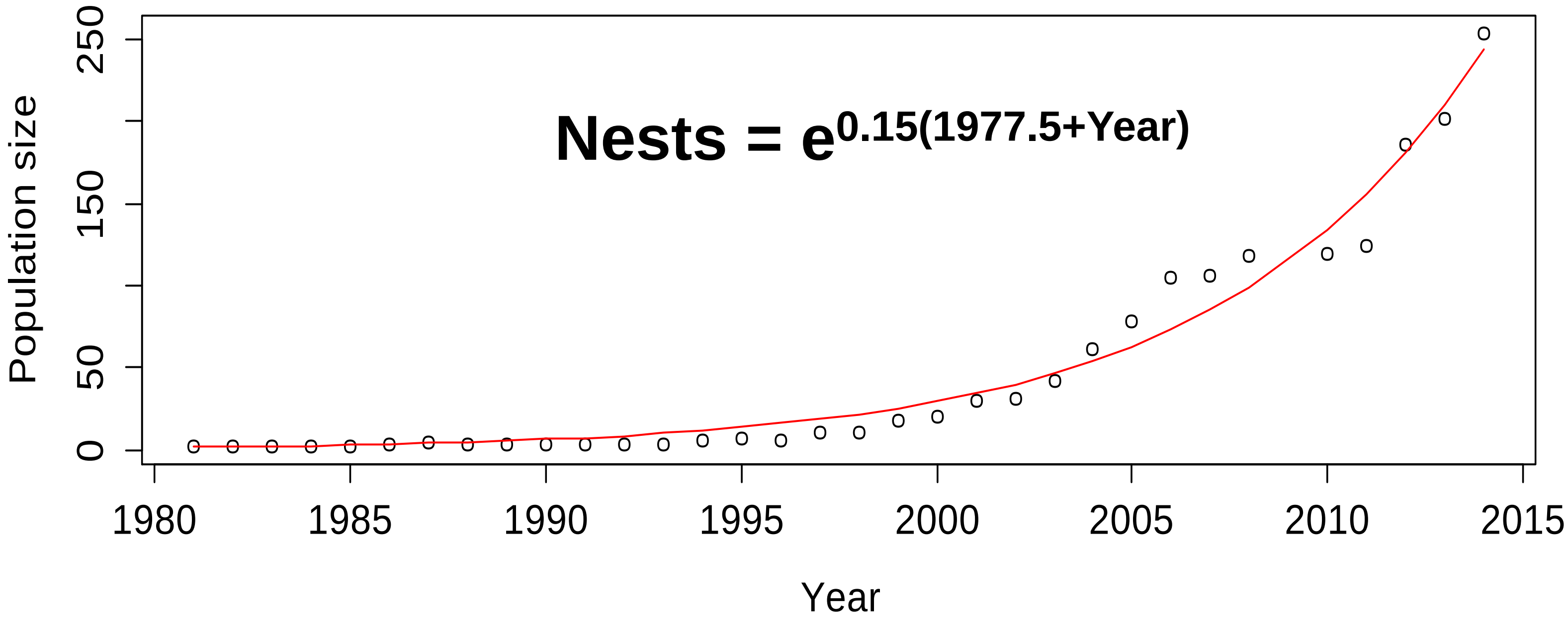




	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.0000667	0.0004349	-0.153	0.882
log(distance)	1.5002315	0.0002077	7222.818	<2e-16 ***

Residual standard error: 0.001016 on 8 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 5.217e+07 on 1 and 8 DF, p-value: < 2.2e-16

Exponential function



Year	Nests
1981	2
1982	2
1983	2
1984	2
1985	2
1986	3
1987	5
1988	3
1989	3
1990	3
1991	3
1992	4
1993	3
1994	6
1995	7
1996	6
1997	11
1998	11
1999	18
2000	20
2001	30
2002	31
2003	42
2004	62
2005	78
2006	105
2007	106
2008	118
2010	119
2011	124
2012	186
2013	201
2014	254

```
out = nls(Nests ~ exp(b1*(b0+Year)),
        data=D, start=list(b0=-1981, b1=1),
        trace = TRUE)
plot(D$Year, D$Nests)
lines(D$Year, fitted(out), col=2)
```

model: Nests ~ exp(b1 * (b0 + Year))

data: D

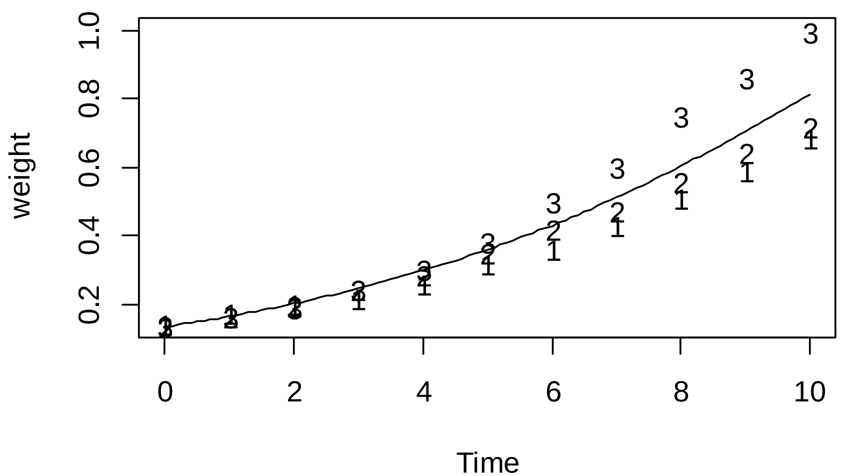
b0 = -1977.5; b1 = 0.15

residual sum-of-squares: 4279

Logistic function

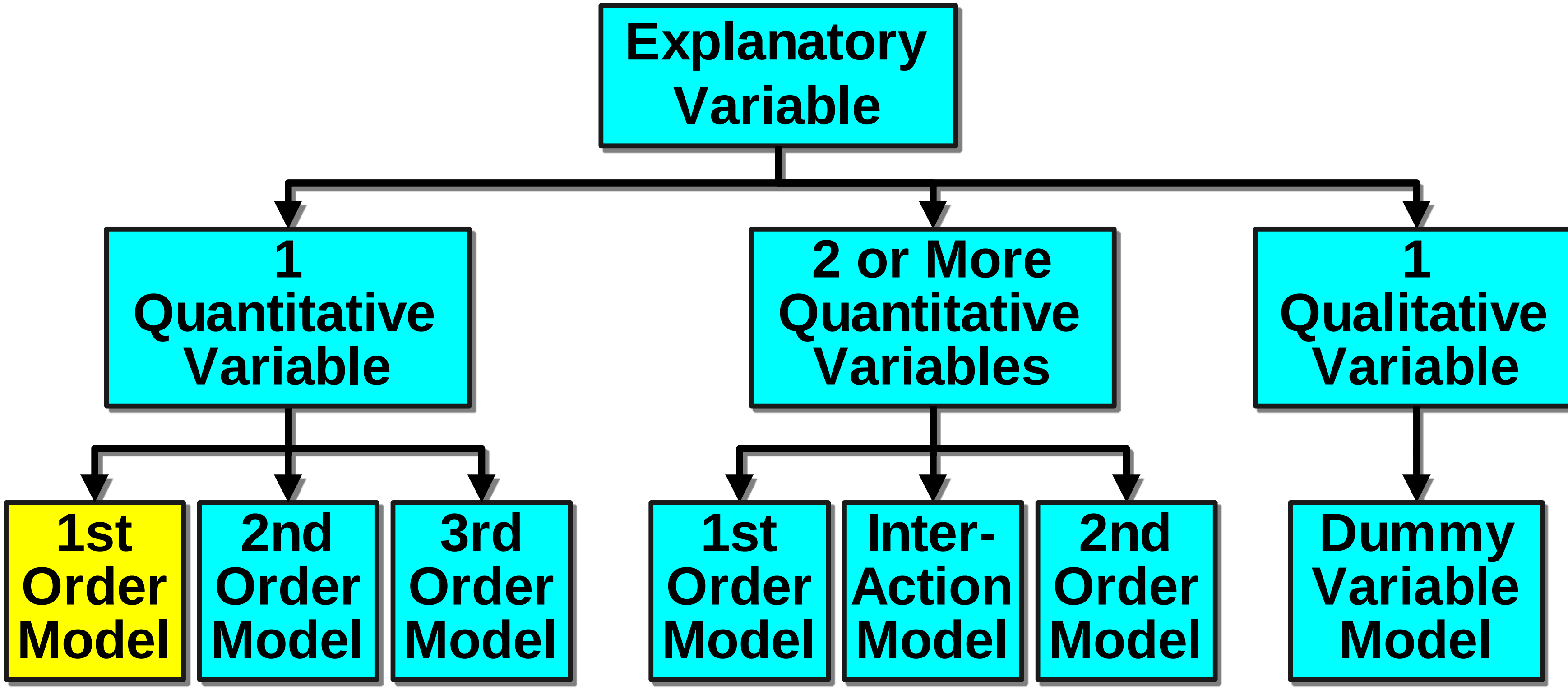
```
# Logistic growth
time <- c(seq(0,10),seq(0,10),seq(0,10))
plant <- c(rep(1,11),rep(2,11),rep(3,11))

weight <- c(
  42,51,59,64,76,93,106,125,149,171,199,
  40,49,58,72,84,103,122,138,162,187,209,

41,49,57,71,89,112,146,174,218,250,288)/288
D <- data.frame(cbind(time, plant, weight))

## Plot weight versus time
plot(
  D$time,
  D$weight,
  xlab="Time",
  ylab="weight",
  type="n"
)

text(
  D$time,
  D$weight,
  D$plant
)
title(main="Graph of weight vs time")
```

$$y = \frac{\alpha}{1 + e^{\beta - \gamma x}}$$




```
IN = getInitial(
  weight ~ SSlogis(time, alpha, xmid, scale),
  data = D
)
## Using the initial parameters above,
## fit the data with a logistic curve.
para0.st <- c(
  alpha   = IN[1],
  beta    = IN[2]/IN[3], # beta is xmid/scale
  gamma= 1/IN[3] # gamma (or r) is 1/scale
)
names(para0.st) = c('alpha', 'beta', 'gamma')

fit0 <- nls(
  weight ~ alpha/(1+exp(beta-gamma*time)),
  D,
  start = para0.st,
  trace = T
)
curve(
  2.21/(1 + exp(2.74 - 0.22*x)),
  from = time[1],
  to = time[11],
  add = TRUE
)
```

Types of regression models (polynomial)



First-order model with 1 independent variable

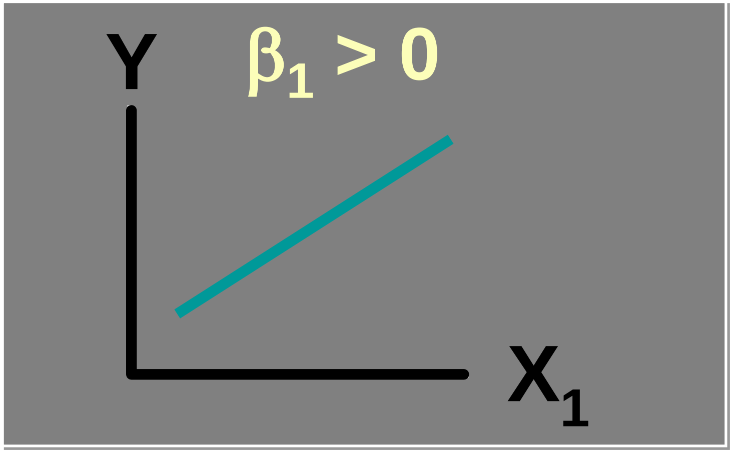
1. Relationship between 1 dependent & 1 independent variable is linear

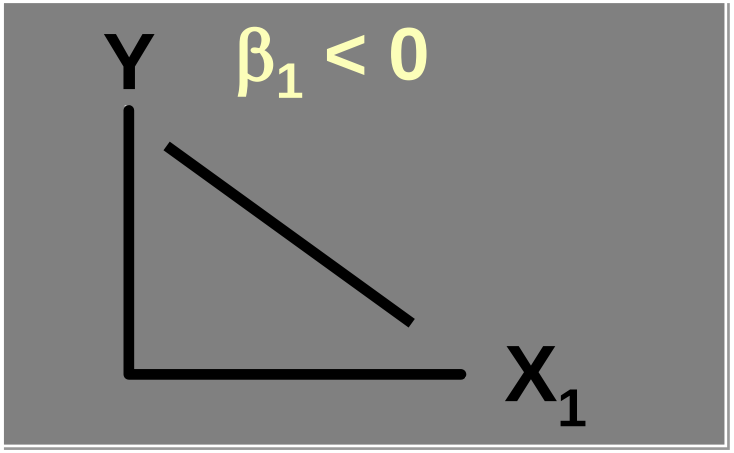
$$E(Y) = \beta_0 + \beta_1 X_{1i}$$

2. Used when expected rate of change in *Y* per unit change in *X* is stable

First-order model relationships

$$E(Y) = \beta_0 + \beta_1 X_1$$



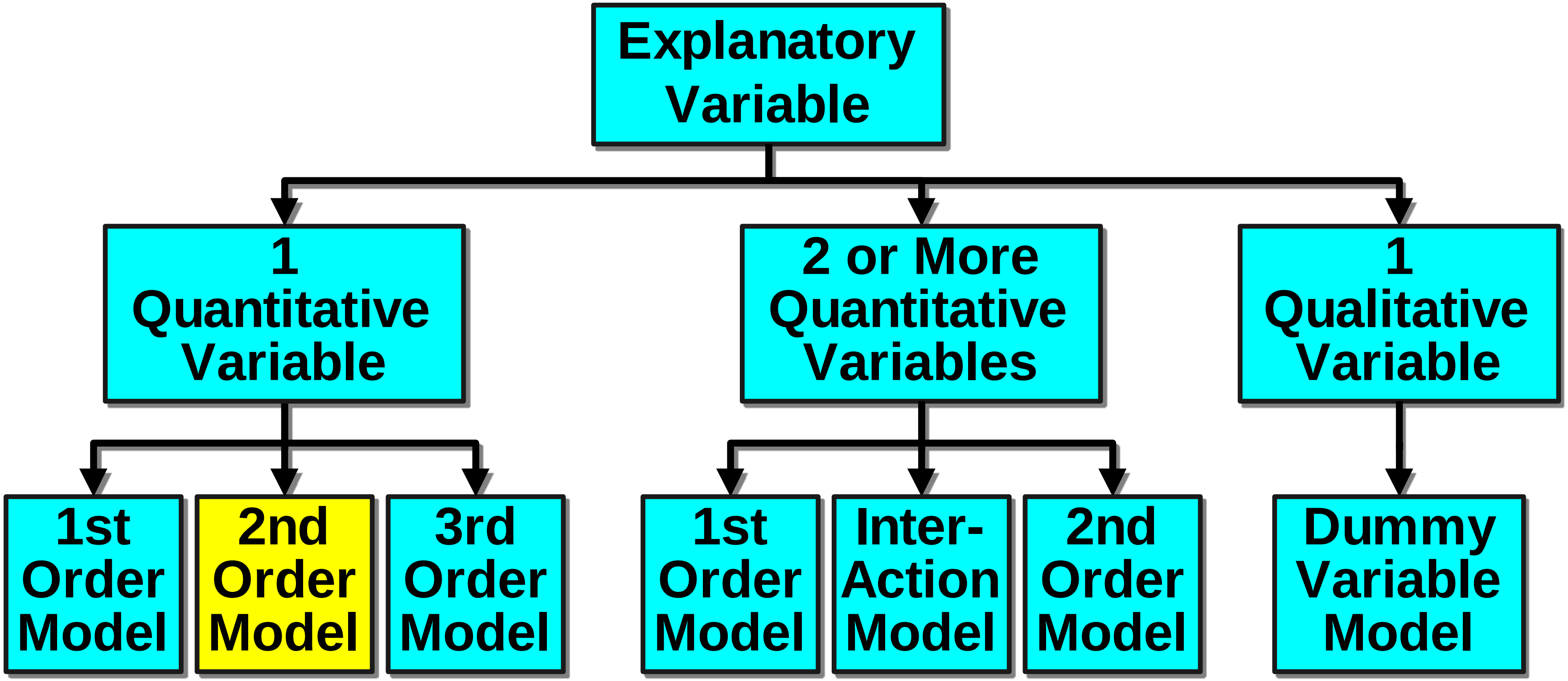


First-order model worksheet

Case, i	Y_i	X_{1i}
1	1	1
2	4	8
3	1	3
4	3	5
:	:	:

Run regression with Y, X_1

Types of regression models (polynomial)



Second-order model with 1 independent variable

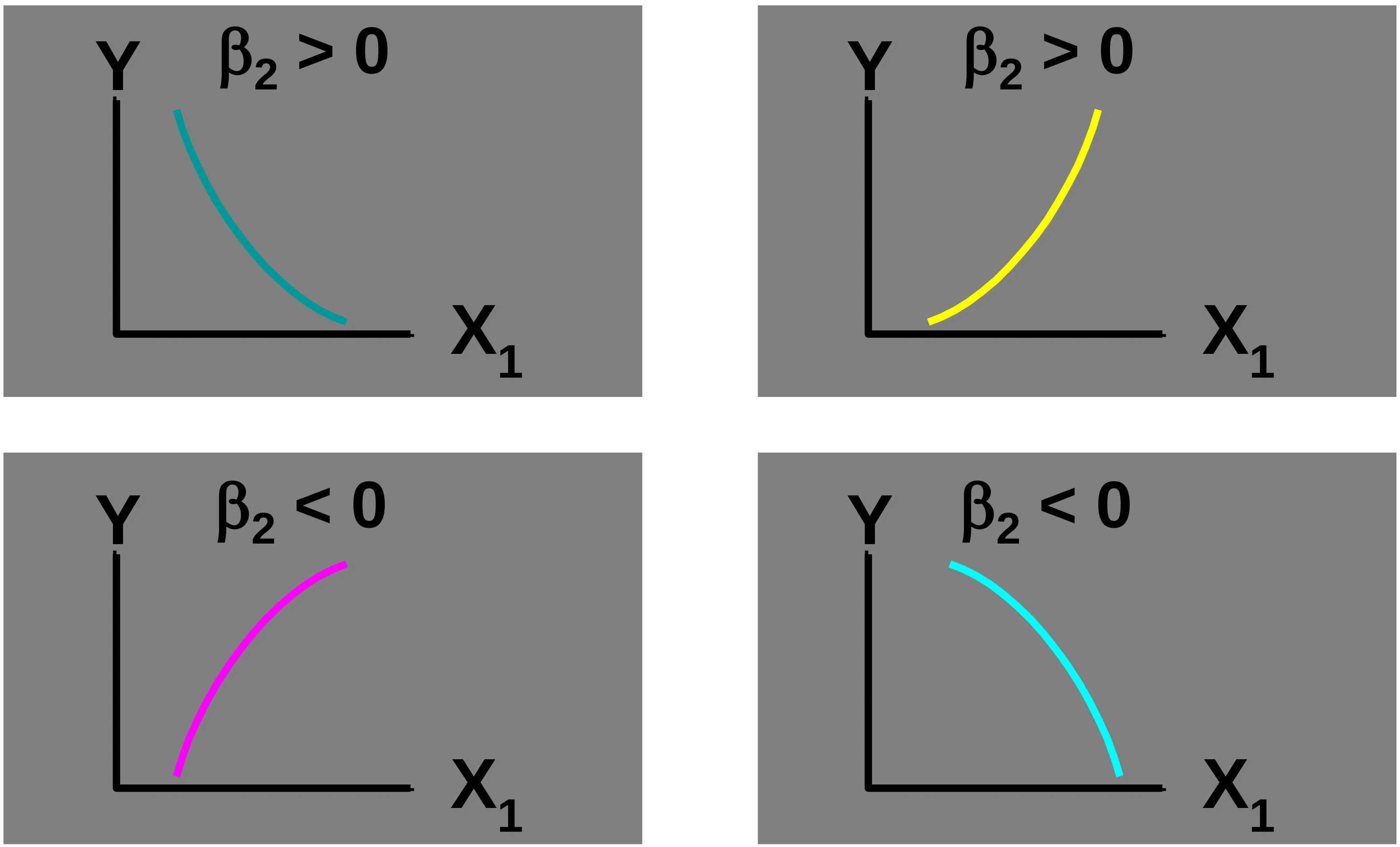
1. Relationship between 1 dependent & 1 independent variables is a quadratic function
2. Model:

Curvilinear effect

$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^{\ 2}$$

Linear effect

Second-order model relationships

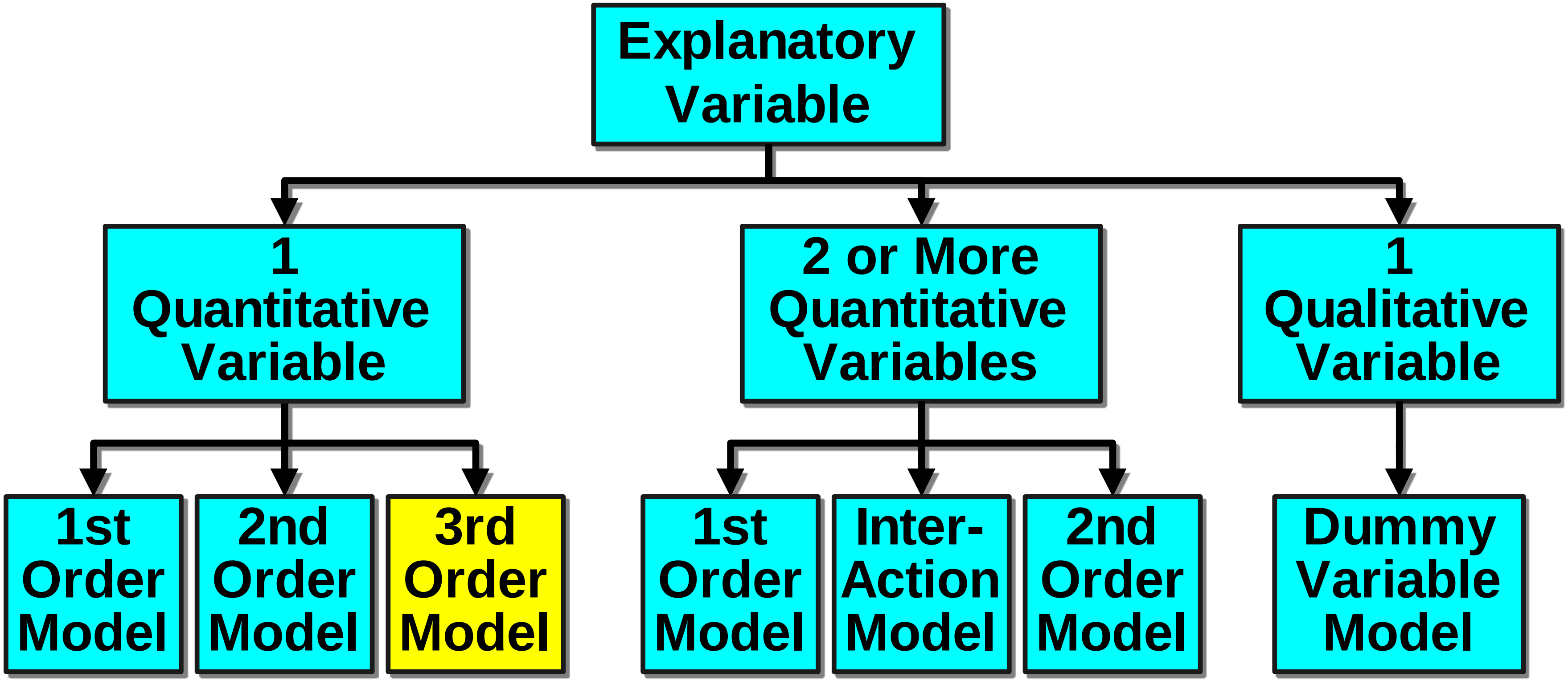


Second-order model worksheet

Case, i	Y_i	X_{1i}	X_{1i}^2
1	1	1	1
2	4	8	64
3	1	3	9
4	3	5	25
:	:	:	:

Create X_1^2 column.
Run linear regression with Y, X_1, X_1^2.

Types of regression models (polynomial)



Third-order model with 1 independent variable

1. Relationship between 1 dependent & 1 independent variable has a ‘wave’
2. Used if 1 reversal in curvature
3. Model:

$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^{\ 2} + \beta_3 X_1^{\ 3}$$

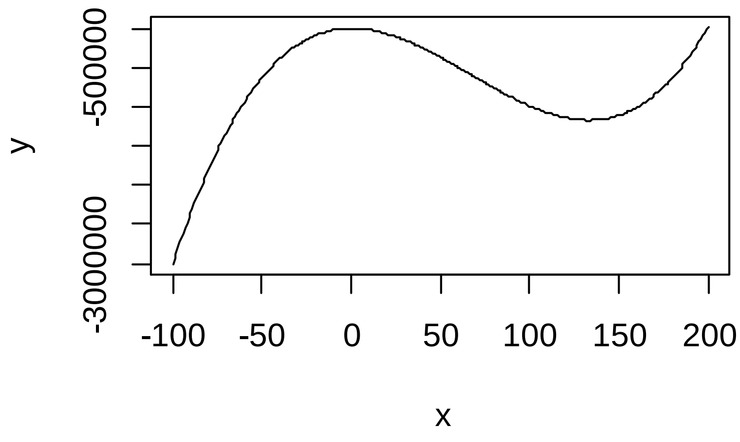
Linear effect

Curvilinear effects

Third-order model relationships

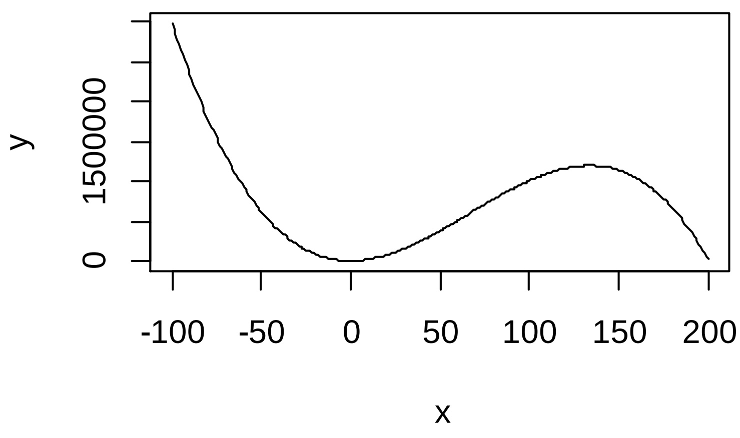
$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^{\ 2} + \beta_3 X_1^{\ 3}$$

$\beta_3 > 0$



y = x^3 - 200*x^2 + 100*x

$\beta_3 < 0$



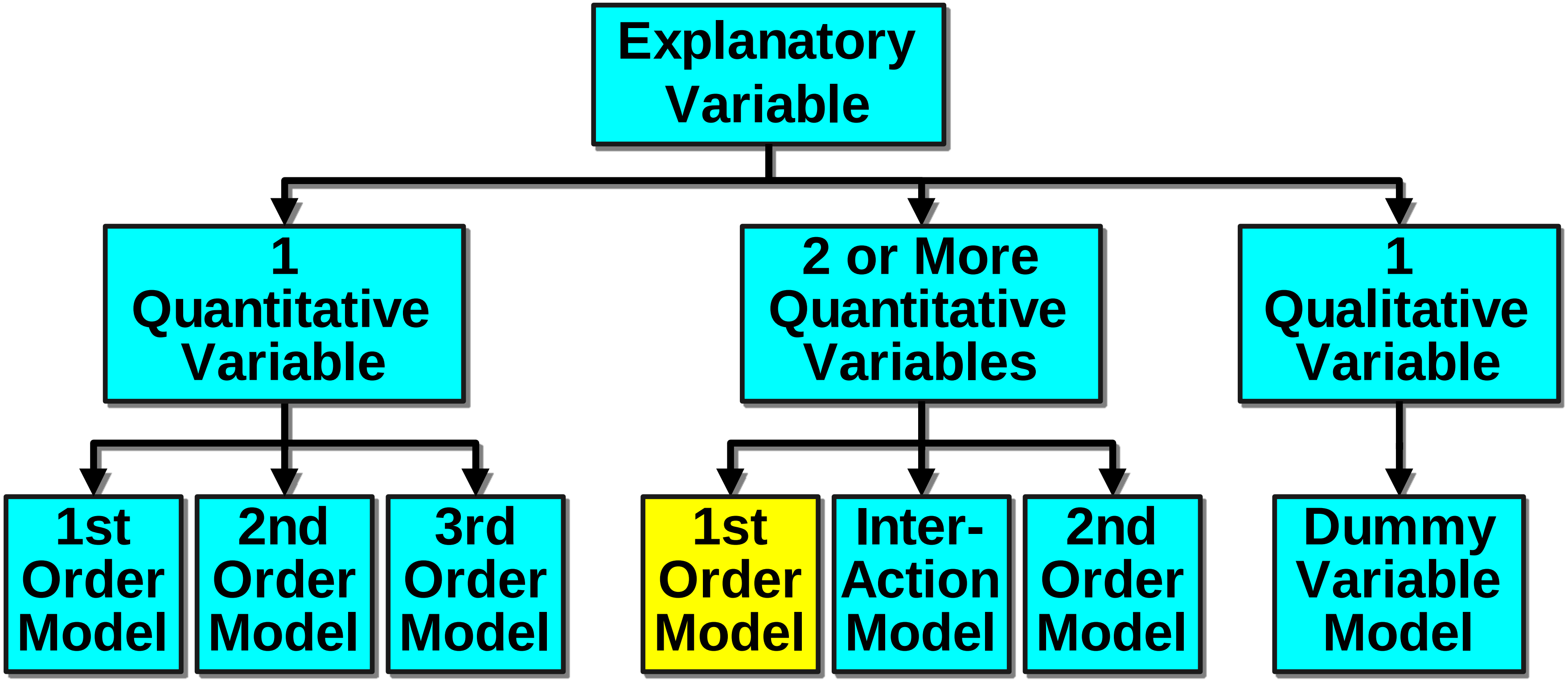
y = -x^3 + 200*x^2 + 100*x

Third-order model worksheet

Case, i	Y_i	X_{1i}	X_{1i}^2	X_{1i}^3
1	1	1	1	1
2	4	8	64	512
3	1	3	9	27
4	3	5	25	125
:	:	:	:	:

Multiply X_1 by X_1 to get X_1^2
Multiply X_1 by X_1 by X_1 to get X_1^3
Run regression with Y, X_1, X_1^2, X_1^3

Types of regression models (polynomial)

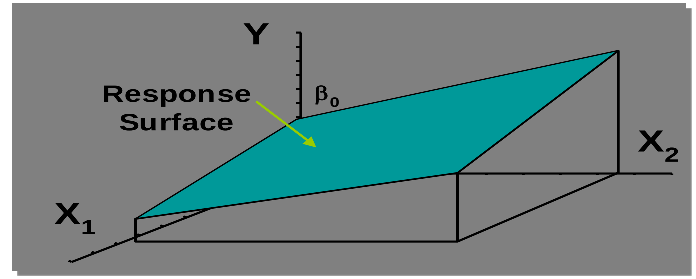


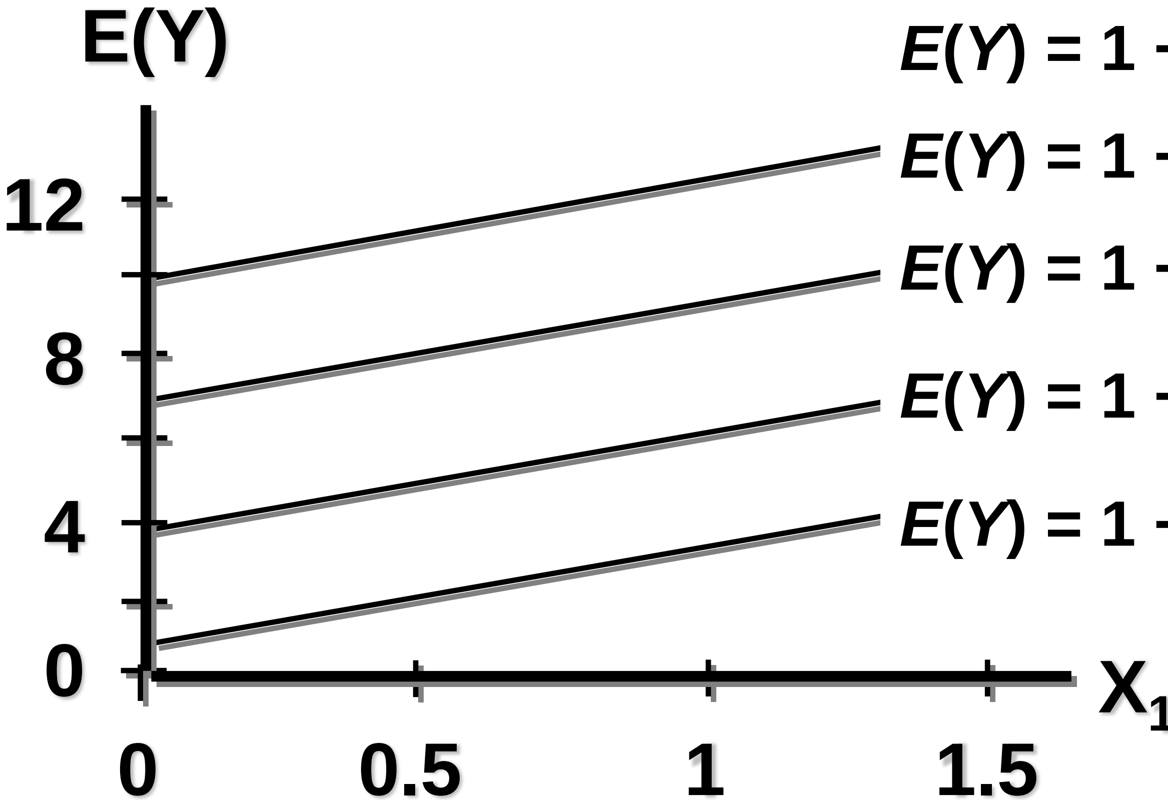
First-order model with 2 independent variables

1. Relationship between 1 dependent & 2 independent variables is a linear function
2. Assumes no interaction between X_1 & X_2
   - Effect of X_1 on $E(Y)$ is the same regardless of X_2 values
3. Model

$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

No interaction





$E(Y) = 1 + 2X_1 + 3X_2$

$E(Y) = 1 + 2X_1 + 3(3) = 10 + 2X_1$

$E(Y) = 1 + 2X_1 + 3(2) = 7 + 2X_1$

$E(Y) = 1 + 2X_1 + 3(1) = 4 + 2X_1$

$E(Y) = 1 + 2X_1 + 3(0) = 1 + 2X_1$

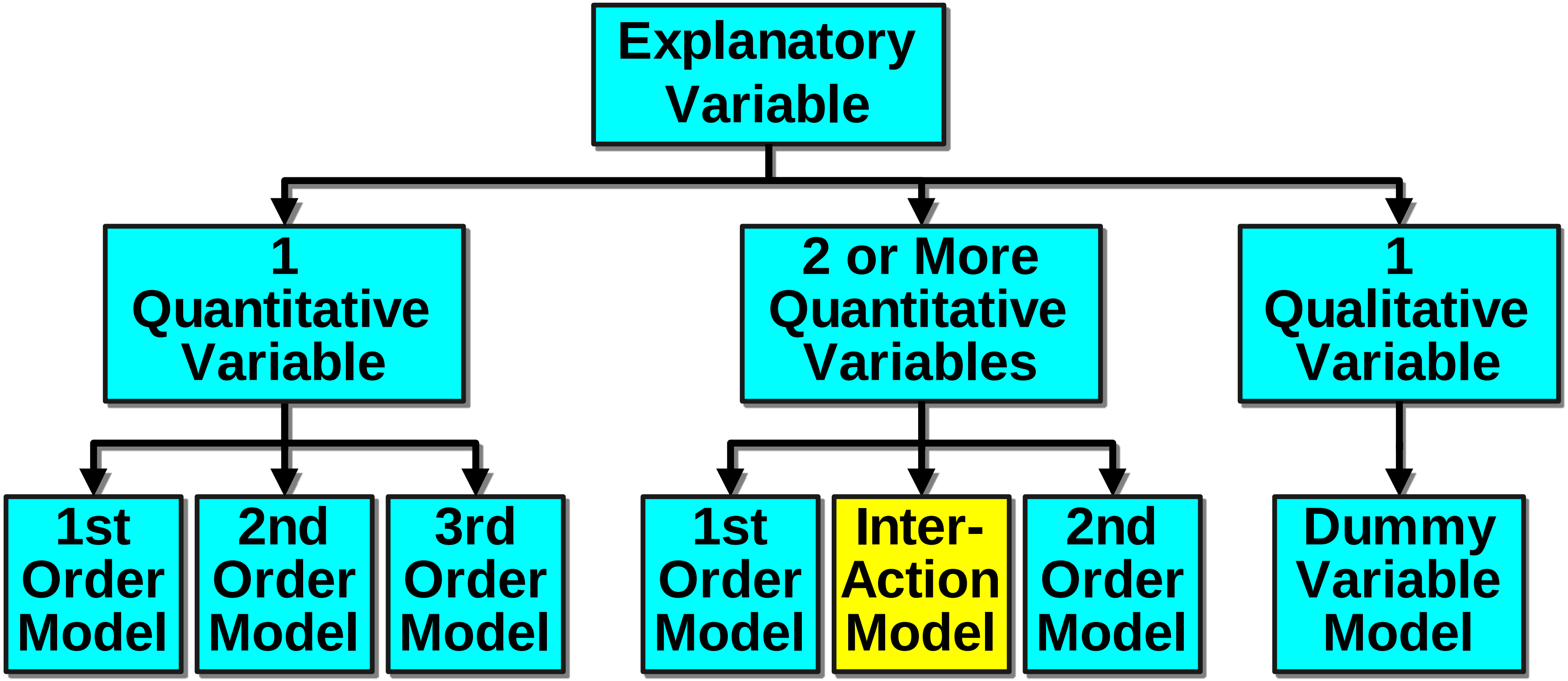
Effect (slope) of X_1 on $E(Y)$ does not depend on X_2 value

First-order model worksheet

Case, i	Y_i	X_{1i}	X_{2i}
1	1	1	3
2	4	8	5
3	1	3	2
4	3	5	6
:	:	:	:

Run regression with Y, X_1, X_2

Types of regression models (polynomial)



Interaction model with 2 independent variables

1. Hypothesizes interaction between pairs of *X* variables

   Response to one *X* variable varies at different levels of another *X* variable

2. Contains two-way cross product terms

$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

Effect of interaction

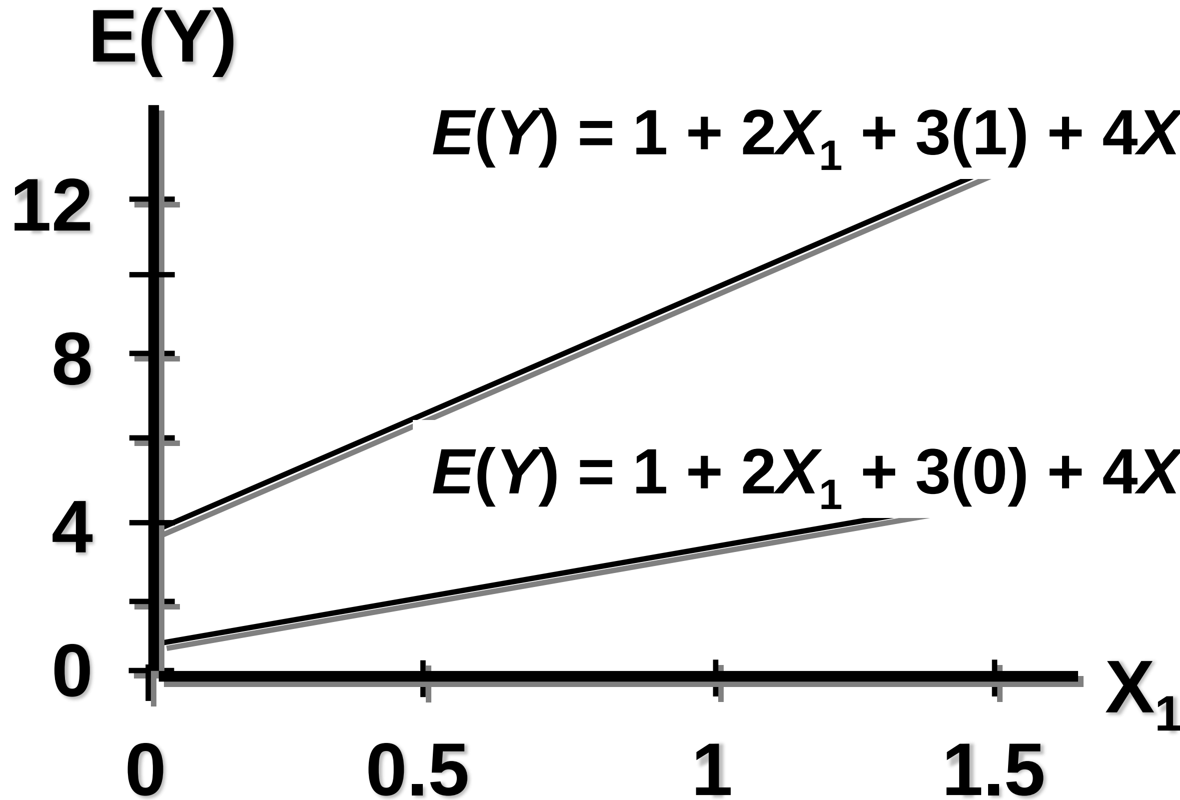
1. Given:

$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2$$

2. Without interaction term, effect of X_1 on Y is measured by β_1
3. With interaction term, effect of X_1 on Y is measured by $\beta_1 + \beta_3 X_2$
   - Effect increases as X_{2i} increases

Interaction model relationships

$E(Y) = 1 + 2X_1 + 3X_2 + 4X_1X_2$



$E(Y) = 1 + 2X_1 + 3(1) + 4X_1(1) = 4 + 6X_1$

$E(Y) = 1 + 2X_1 + 3(0) + 4X_1(0) = 1 + 2X_1$

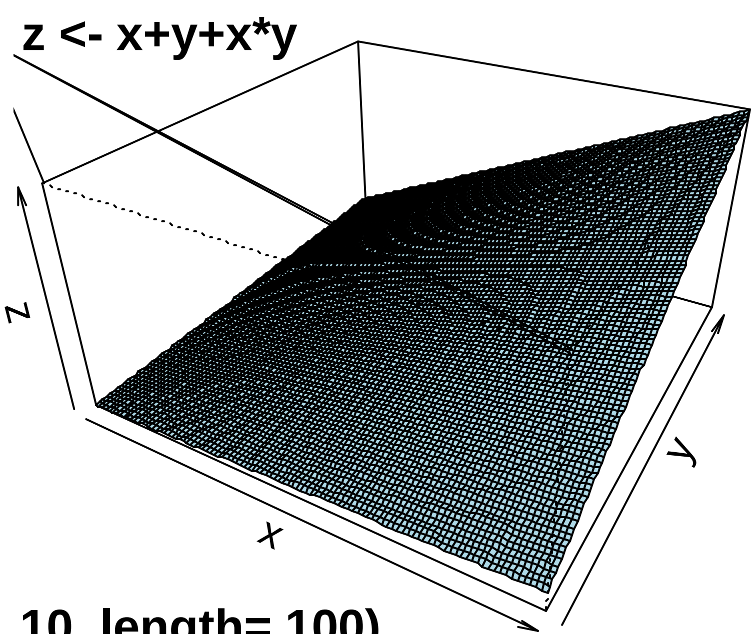
Effect (slope) of X_1 on $E(Y)$ does depend on X_2 value

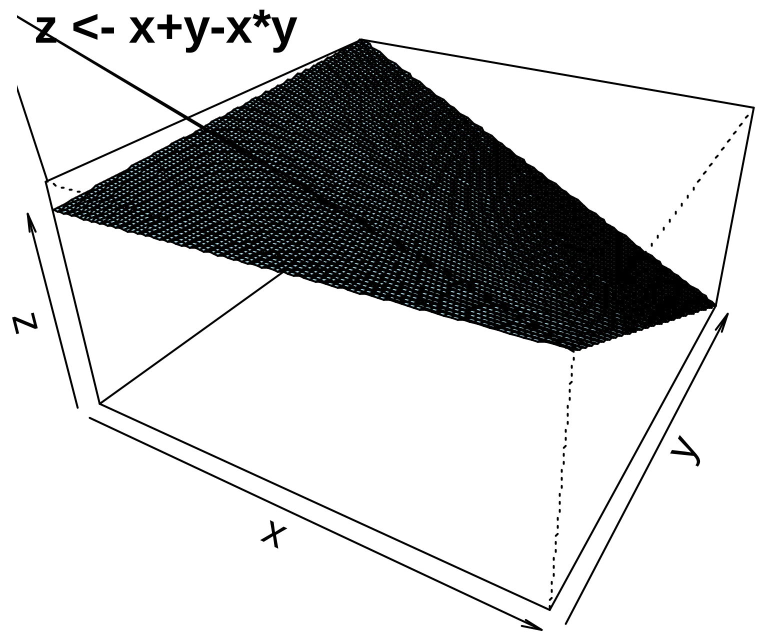
Interaction model worksheet

Case, i	Y_i	X_{1i}	X_{2i}	$X_{1i}\,X_{2i}$
1	1	1	3	3
2	4	8	5	40
3	1	3	2	6
4	3	5	6	30
:	:	:	:	:

Multiply X_1 by X_2 to get X_1X_2.
Run regression with Y, X_1, X_2 , X_1X_2

Perspective plots for interaction models

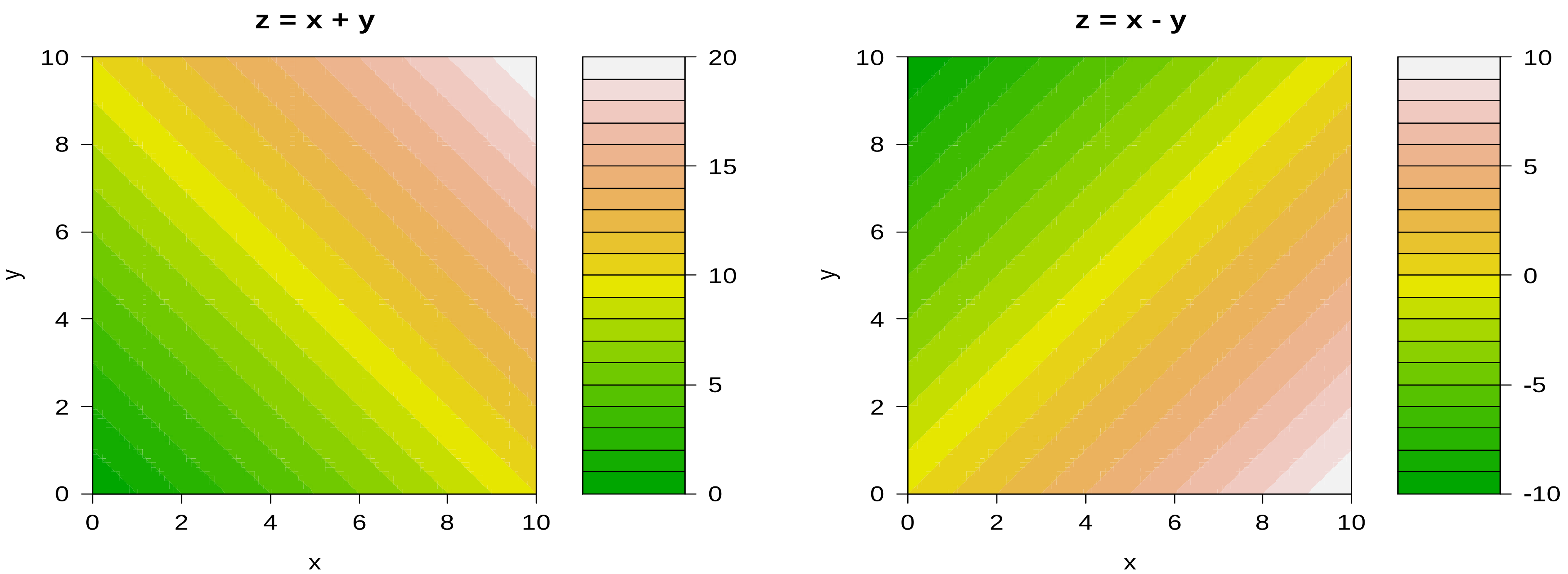





```
x <- seq(0, 10, length= 100)
y <- x
f <- function(x, y) { r <- x+y+x*y }
z <- outer(x, y, f)

op <- par(bg = "white", mfrow=c(1,2))
persp(x, y, z, theta = 30, phi = 30, expand = 0.5,
      col = "lightblue", main='z=x+y+x*y')
```

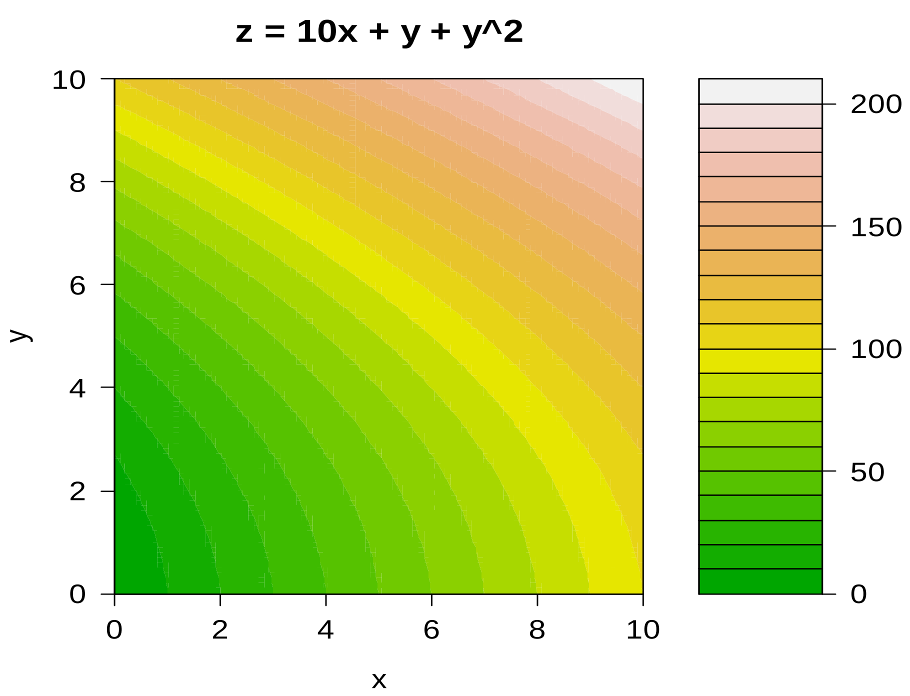
Contour plots for models with linear terms

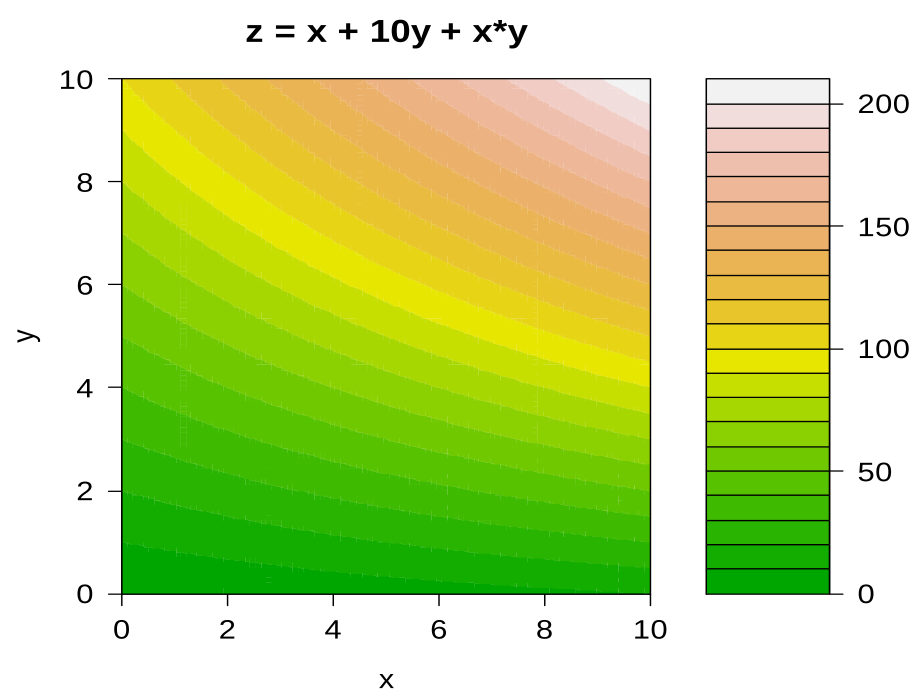
```
x = y <- seq(0, 10, length= 100); f <- function(x, y) { r <- x+y }; z <- outer(x, y, f)
    filled.contour(x, y, z, main="z = x + y", color = terrain.colors)
```



Contour plots for high order models

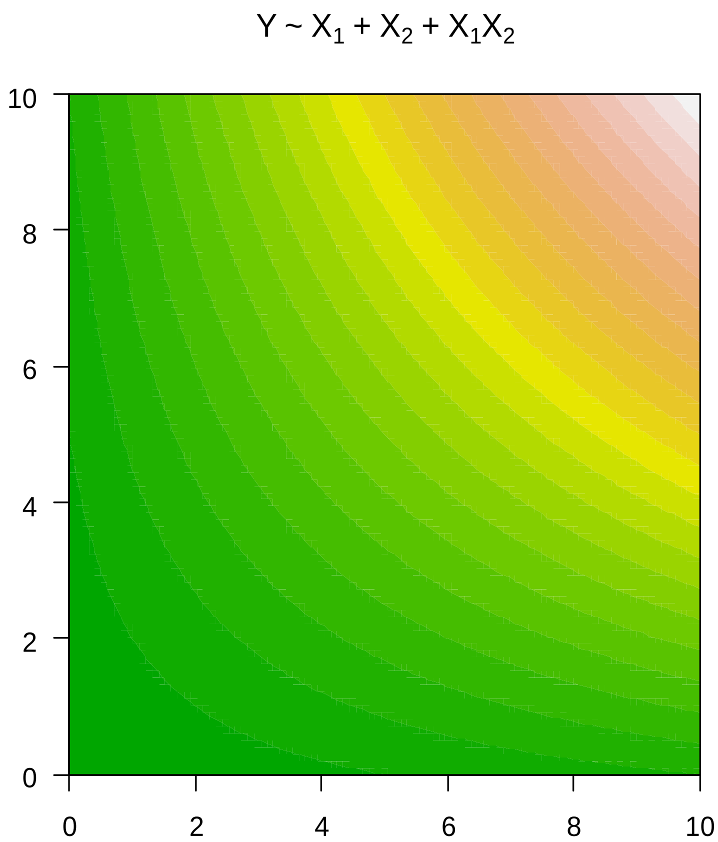
filled.contour(x, y, z, main="z = x + 10y + xy", color = terrain.colors)

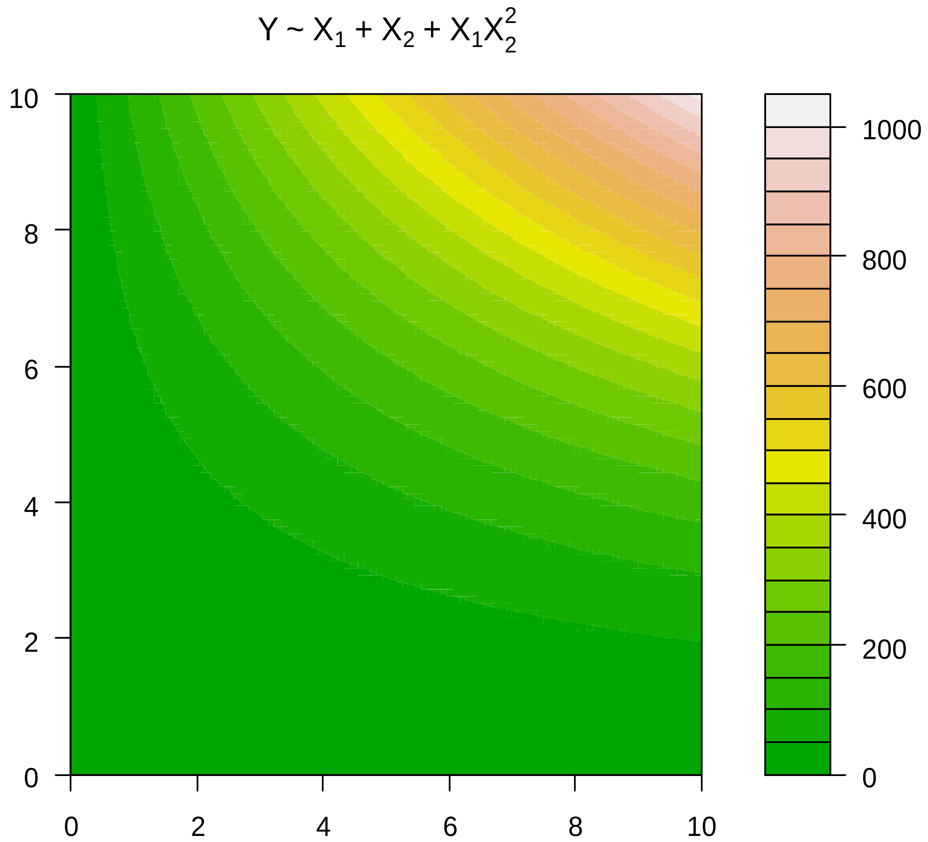




Contour plots for interaction models

```
x1 = x2 <- seq(0, 10, length= 100); f <- function(x1, x2) { r <- x1+x2+x1*x2*x2 }; y <- outer(x1, x2, f)
filled.contour(x1, x2, y, main=expression(paste("Y ~ ", X[1], " + ", X[2], " + ", X[1], X[2]^2)), color = terrain.colors)
# [] subscript; ^ superscript
```

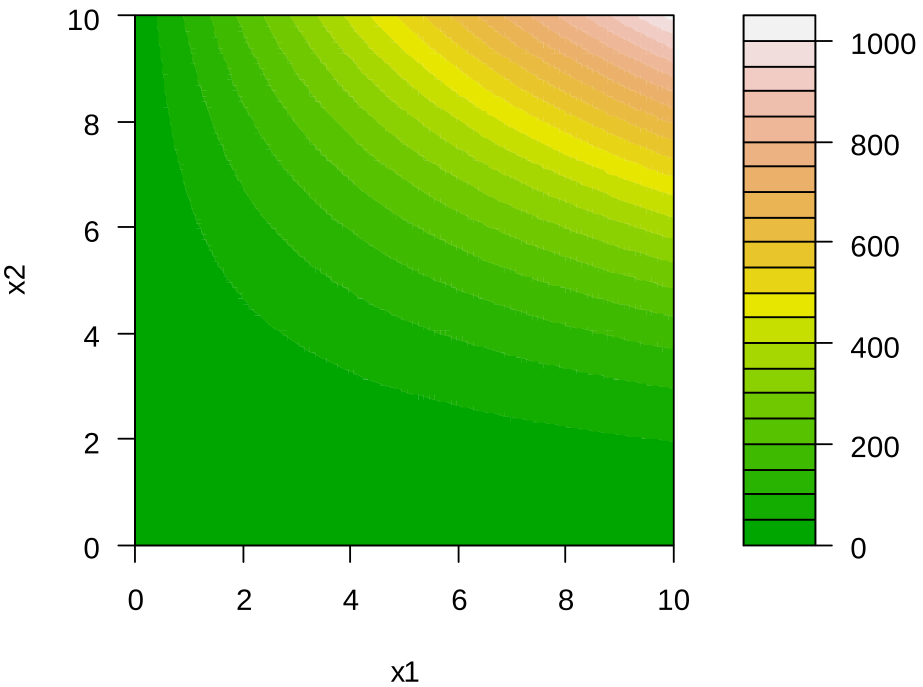




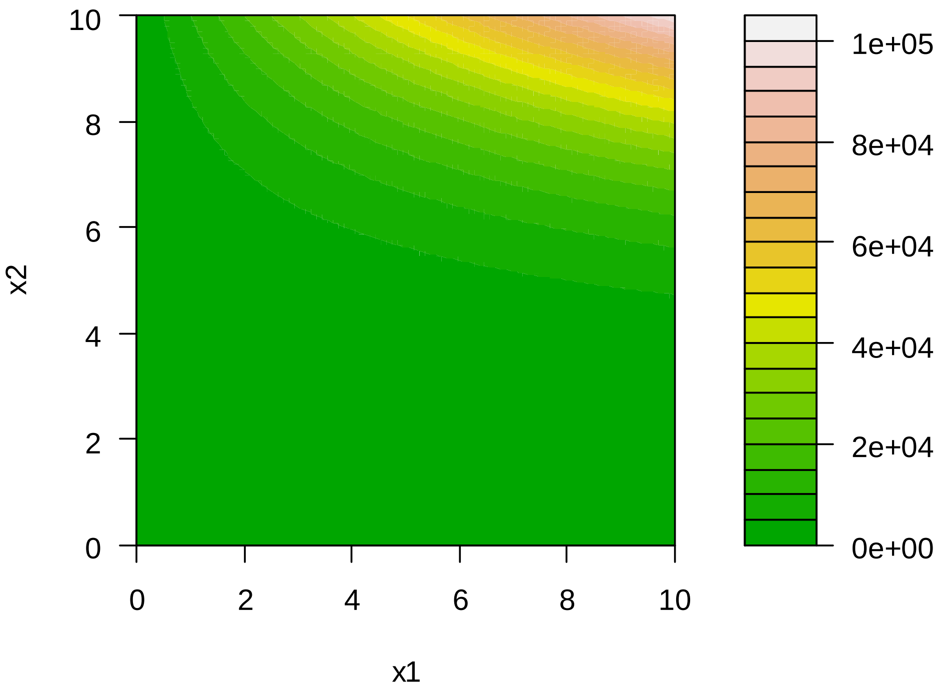
Estimating regression coefficients

x1 = 0.8485 x2 = 1.0099 x1:x2 = 0.9787	x1 = 0.8198 x2 = 1.0145 x1:x2 = 0.7744

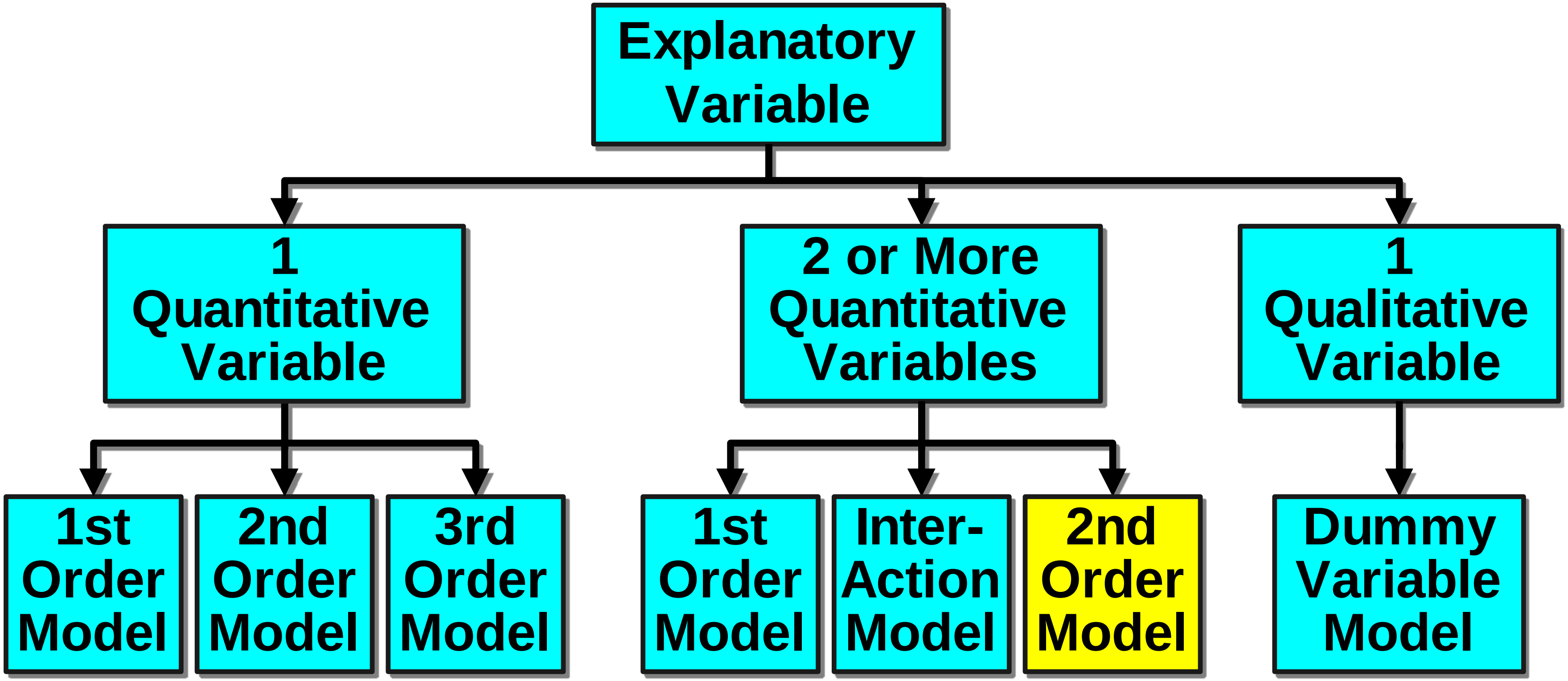
y = x1 + x2 + x1*x2^2



y = x1 + x2 + x1*x2^4



Types of regression models (detailed)



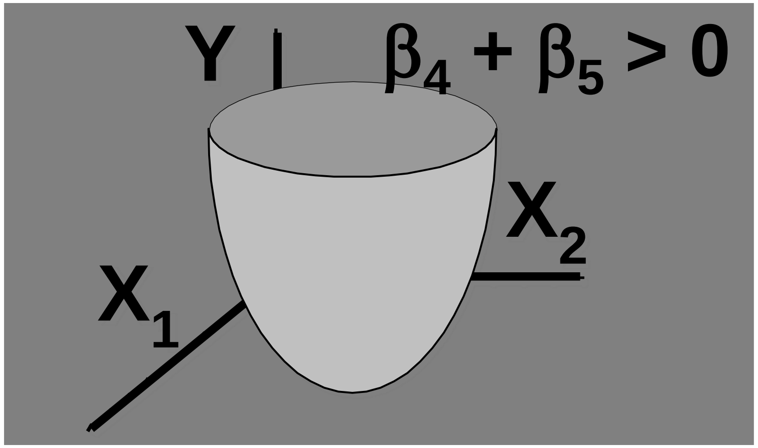
Second-order model with 2 independent variables

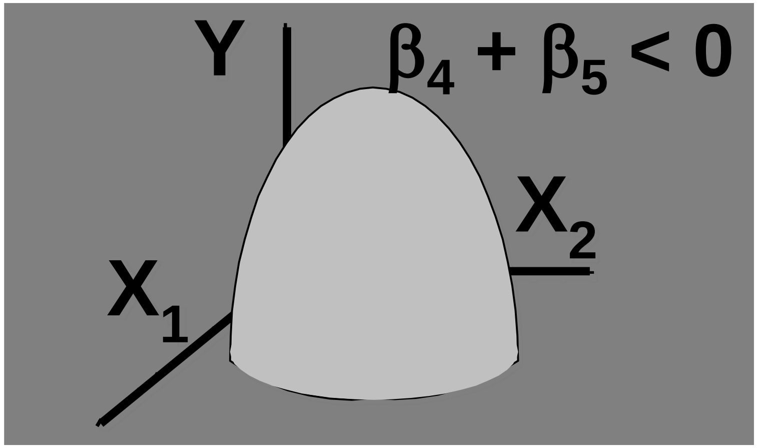
1. Relationship between 1 dependent & 2 or more independent variables is a quadratic function

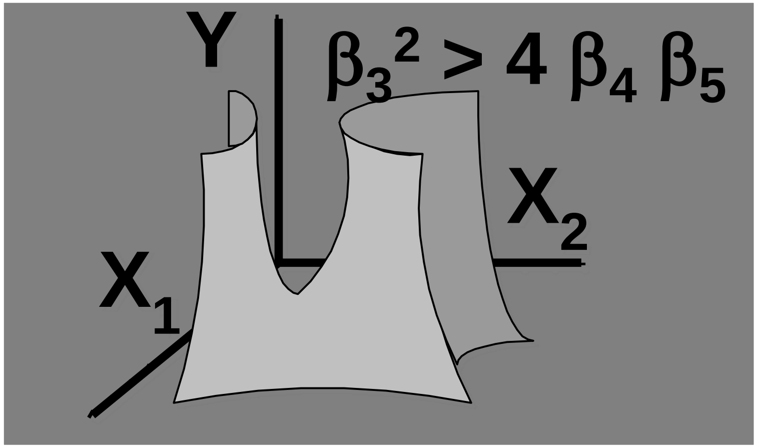
2. Use model:

$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2 + \beta_5 X_2^2$$

Second-order model relationships







$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \beta_4 X_1^2 + \beta_5 X_2^2$$

Second-order model worksheet

Case, i	Y_i	X_{1i}	X_{2i}	$X_{1i}X_{2i}$	X_{1i}^2	X_{2i}^2
1	1	1	3	3	1	9
2	4	8	5	40	64	25
3	1	3	2	6	9	4
4	3	5	6	30	25	36
:	:	:	:	:	:	:

Multiply X_1 by X_2 to get X_1X_2; then X_1^2, X_2^2.
Run regression with Y, X_1, X_2 , X_1X_2, X_1^2, X_2^2.

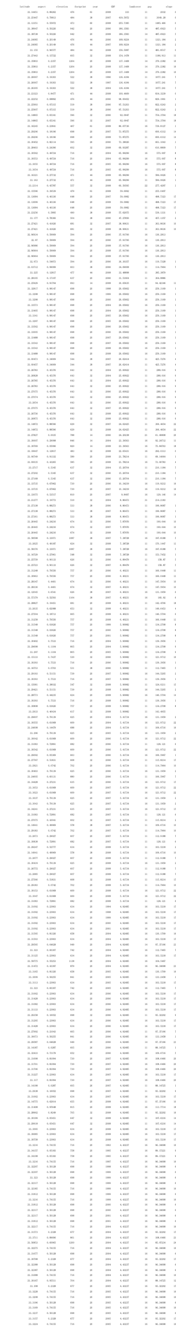
R code - multiple linear regression

```
ibis = read.csv('D:/database/ibisdata/ibis2010.csv', header=T)
head(ibis)
ibis.pre = ibis[ibis$use==1,c(3:6,8,9,11,12)]
head(ibis.pre)
```

	latitude	aspect	elevation	footprint	year	GDP	pop	slope
1	33.1	0.893	476	61	2008	333	2032	0.503
42	33.3	0.798	484	38	2007	420	3049	0.685
86	33.1	0.56	473	60	2008	256	1485	0.812
104	33.4	0.502	942	20	2006	186	488	5.002
105	33.4	0.502	942	20	2008	186	488	5.002
116	33.2	0.201	476	44	2006	169	1321	2.275

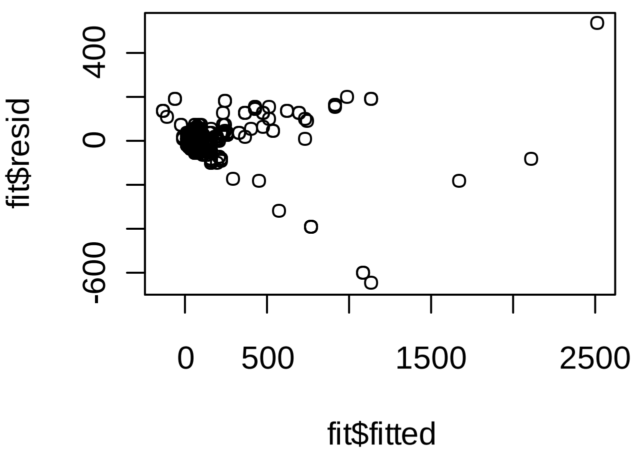
```
# Multiple Linear Regression Example (only include linear terms)
fit <- lm(pop ~ latitude+elevation+footprint+year+GDP+slope, data=ibis.pre)
summary(fit) # show results
```

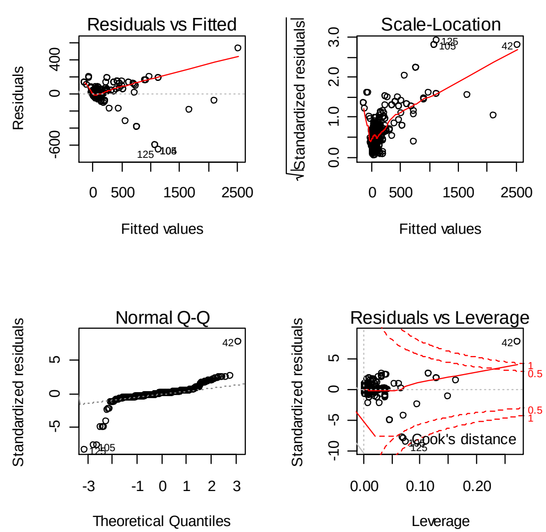
Coefficients:	Estimate	Std.Error	t value	Pr(>|t|)
(Intercept)	-8670.00000	2120.00000	-4.10000	0.00005
latitude	208.00000	49.80000	4.17000	0.00004
elevation	-0.14400	0.01930	-7.47000	0.00000
footprint	4.43000	0.62400	7.10000	0.00000
year	0.90300	0.64300	1.40000	0.16000
GDP	5.63000	0.11200	50.39000	<0.00000
slope	0.65700	0.54100	1.21000	0.23000



R code - multiple linear regression

```
# Other useful functions
coefficients(fit) # model coefficients
confint(fit, level=0.95) # CIs for model parameters
fitted(fit) # predicted values
residuals(fit) # residuals
anova(fit) # anova table
vcov(fit) # covariance matrix for model parameters

# diagnostic plots
plot(fit$fitted, fit$resid)
layout(matrix(c(1,2,3,4),2,2)) # optional 4 graphs
plot(fit)
```





R code - multiple linear regression

```
> step <- stepAIC(fit, direction="both")
Start:  AIC=4658
pop ~ y + elevation + footprint + year + GDP + slope

            Df Sum of Sq      RSS  AIC
- slope      1      9244  3300402 4658
- year       1     12344  3303502 4658
<none>                    3291158 4658
- y          1    108873  3400031 4674
- footprint  1    316173  3607331 4705
- elevation  1    349906  3641064 4710
- GDP        1  15920259 19211417 5595

Step:  AIC=4658
pop ~ y + elevation + footprint + year + GDP

            Df Sum of Sq      RSS  AIC
- year       1     11643  3312045 4658
<none>                    3300402 4658
+ slope      1      9244  3291158 4658
- y          1    114255  3414656 4674
- footprint  1    306991  3607392 4703
- elevation  1    346676  3647078 4709
- GDP        1  15955393 19255794 5594

Step:  AIC=4658
pop ~ y + elevation + footprint + GDP

            Df Sum of Sq      RSS  AIC
<none>                    3312045 4658
+ year       1     11643  3300402 4658
+ slope      1      8543  3303502 4658
- y          1    112618  3424663 4674
- footprint  1    315040  3627084 4704
- elevation  1    373870  3685915 4713
- GDP        1  16068807 19380852 5596
```

```
# Stepwise Regression
library(MASS)
fit <- lm(pop ~ y+elevation+footprint+year+GDP+slope,
        data=ibis.pre)
step <- stepAIC(fit, direction="both")
step$anova # display results

# use mtcars data
fit <- lm(mpg ~ ., data=mtcars)
```

```
> step$anova # display results
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
pop ~ y + elevation + footprint + year + GDP + slope

Final Model:
pop ~ y + elevation + footprint + GDP


     Step Df Deviance Resid. Df Resid. Dev  AIC
1                           525    3291158 4658
2 - slope  1     9244       526    3300402 4658
3  - year  1    11643       527    3312045 4658
```

Use the full model as a start

```
attach(trees)
fit = lm(Volume ~ Girth * Height + I(Girth^2) + I(Height^2),
         data=trees)
fit = step(fit)
summary(fit)

# A general type
fit = lm(Y ~ (X1 + X2 + X3)^2 + I(X1^2) + I(X2^2) + I(X3^2),
         data=mydata)
```

Assignment

General objectives: learn about multiple linear regression.

- Make a dataset ready, including at least three continuous variables Y, X1 and X2 (X3 and X4 are suggested to be included).

- Check multicollinearity (column relationship) and independence (row relationship).

- Start from the full model, including all quadratic terms and interaction terms

```
fit = lm(Y ~ X1 + X2 + I(X1^2) + I(X2^2) + X1:X2, data=mydata).
```

- Run model selection and remove insignificant variables and terms.

- Report R^2, significance of each variables and terms, homogeneous of residuals.

- Briefly interpret the results.