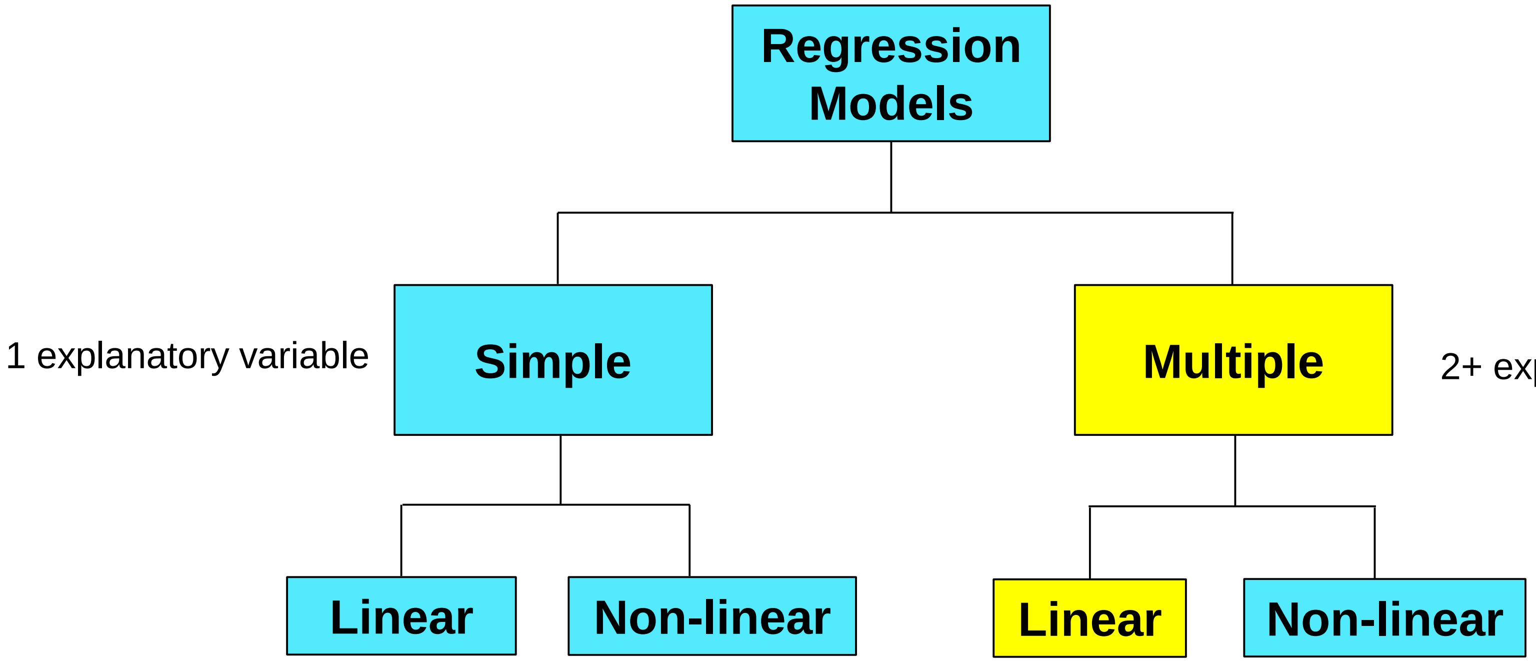
Multiple regression and correlation

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

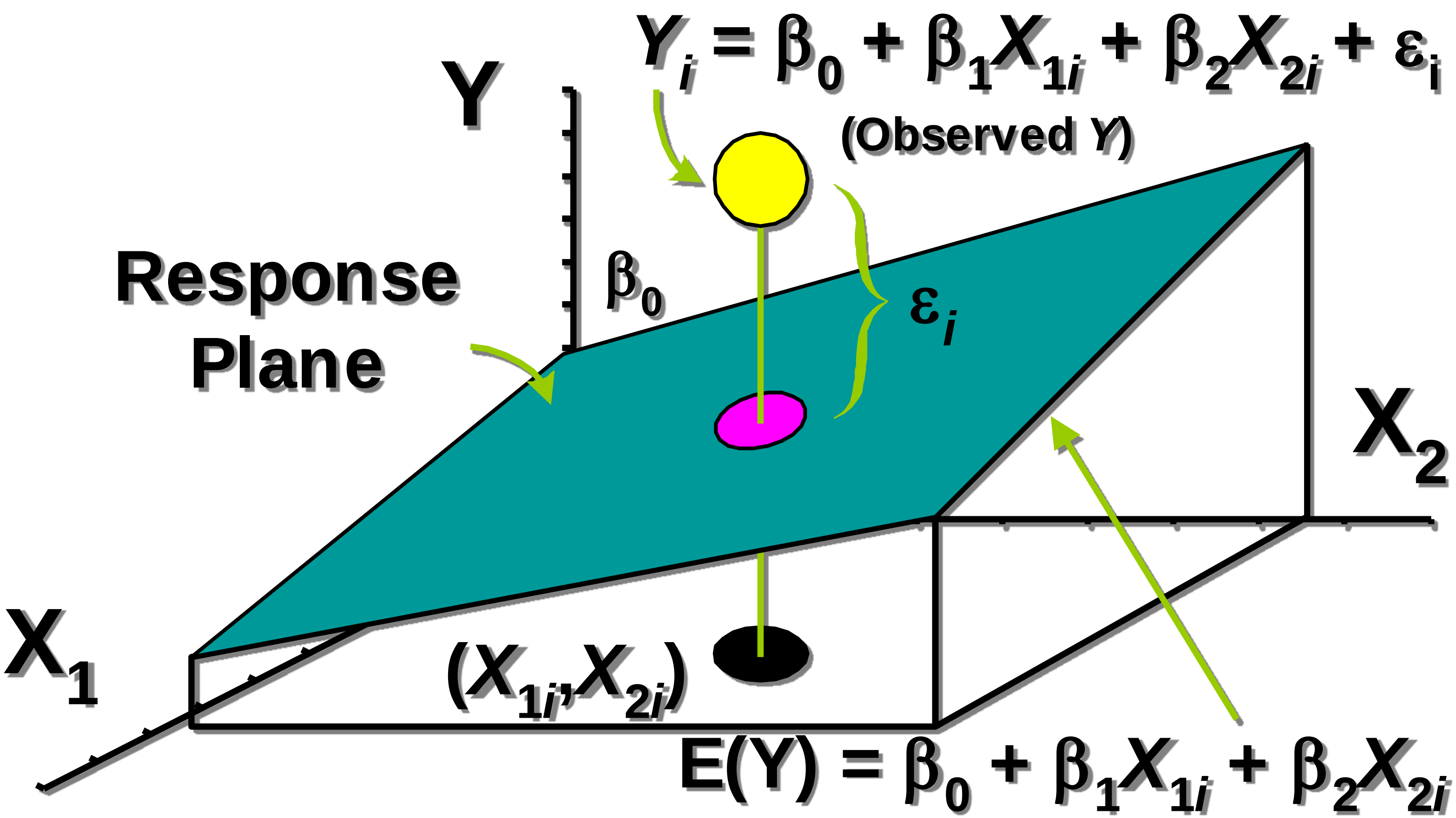
Random error

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

Dependent (response) variable

Independent (explanatory) variables

Bivariate regression model



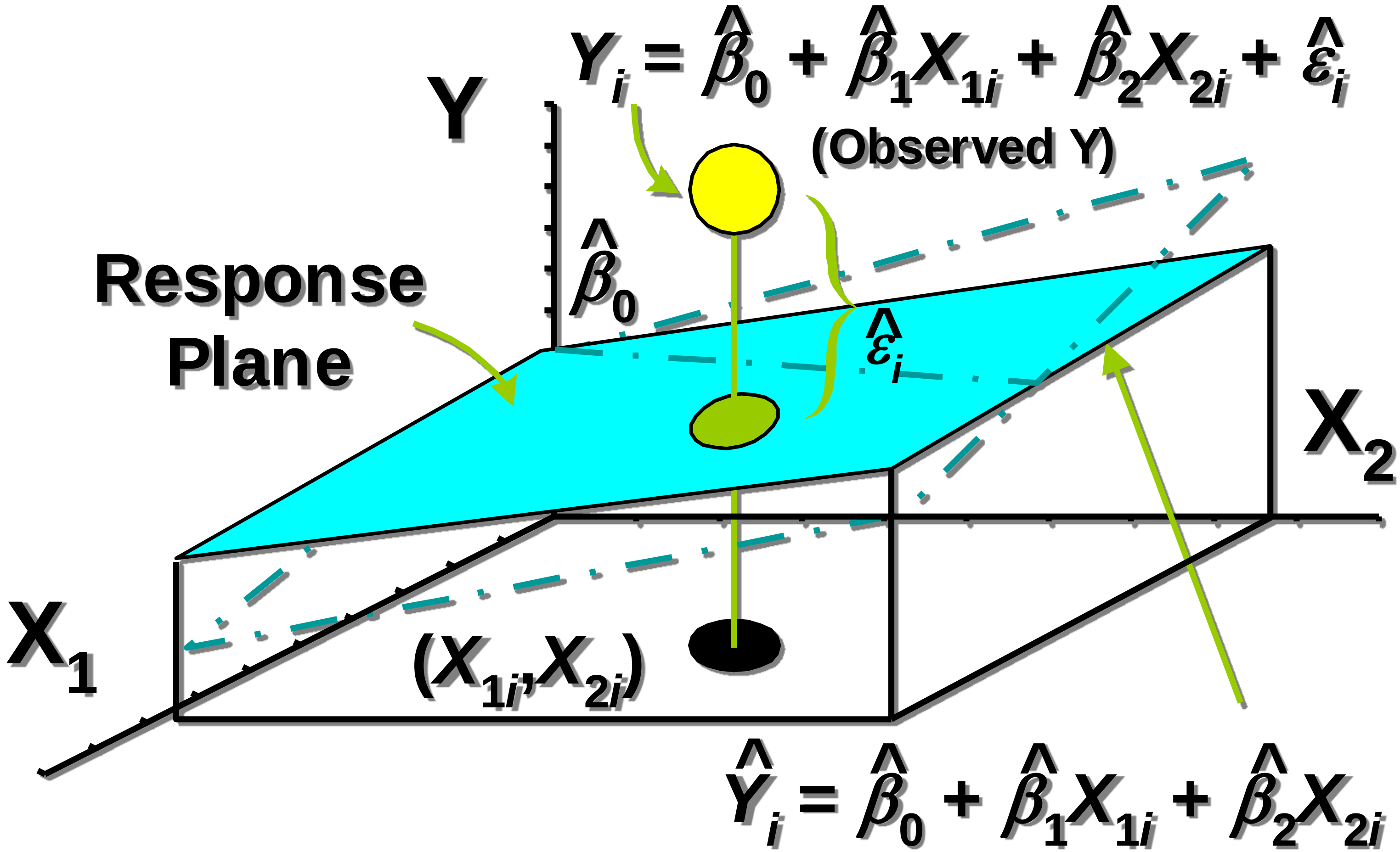
Regression modeling steps

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 :

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

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

In matrix notation :

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

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

X' (X-prime or X-transpose)

Sum of squares and cross-products matrix (SSCP)

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

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

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

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:

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

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 ($\beta_i < 0$)

Two-tailed test

$H_0: \beta_i = 0$

$H_a: \beta_i \neq 0$

test statistic: $t = \dfrac{\hat{\beta}_i}{s\sqrt{c_{ii}}}$

Rejection region:

One-tailed: $t > t_\alpha$ (or $t < t_\alpha$)

Two-tailed: $|t| > t_{\alpha/2}$

$t_{\alpha/2}$ is based on [n - (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 SD (standard deviation)

$$S_{\hat{\beta}_i} = s\sqrt{c_{ii}}$$

Parameter distribution: $t = \dfrac{\hat{\beta}_i}{s\sqrt{c_{ii}}}$

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 Xs.

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

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

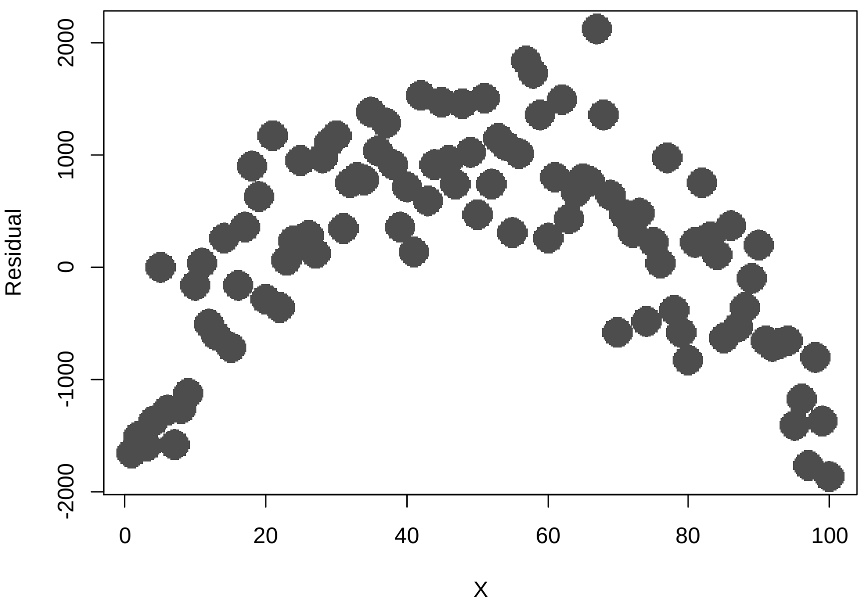
2. Purposes
   - Examine functional form (linear vs. non-linear model)
   - Evaluate violations of assumptions

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

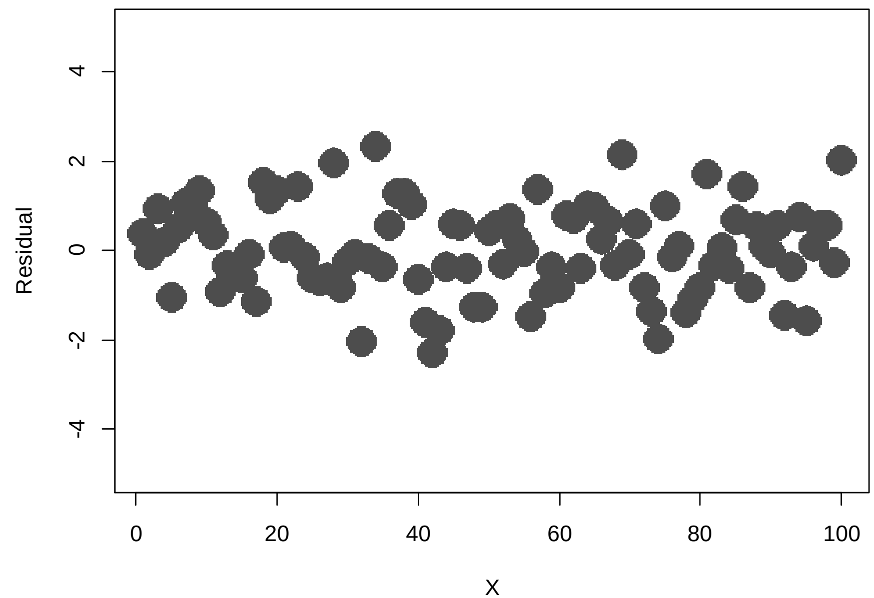
Residual plot for functional form

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')

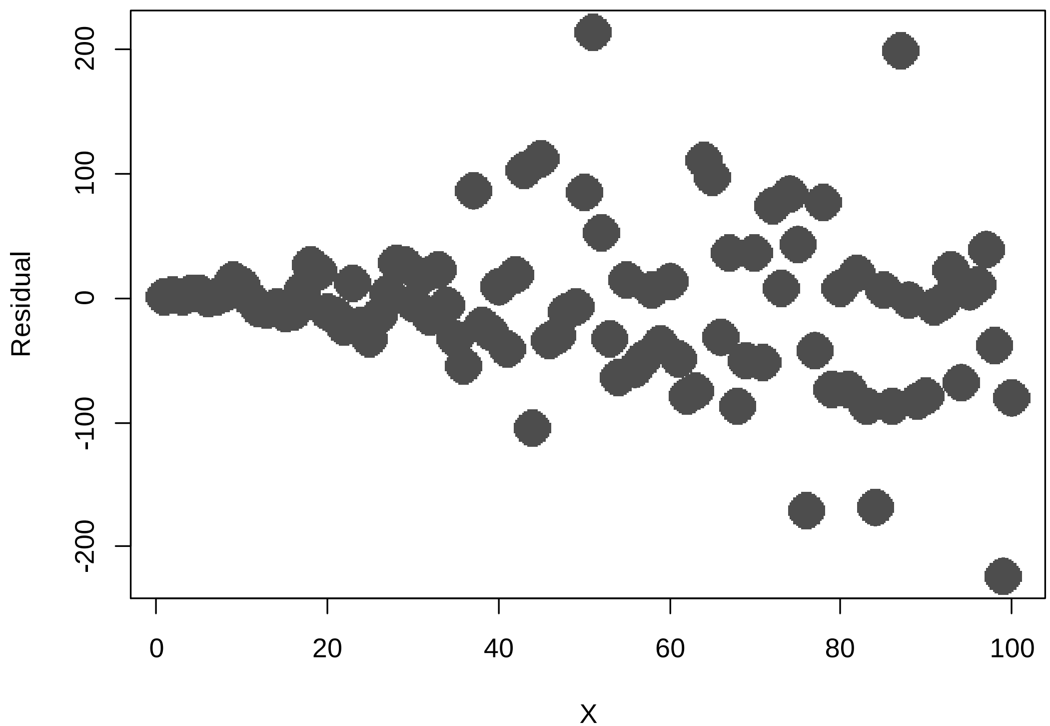
Correct Specification



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

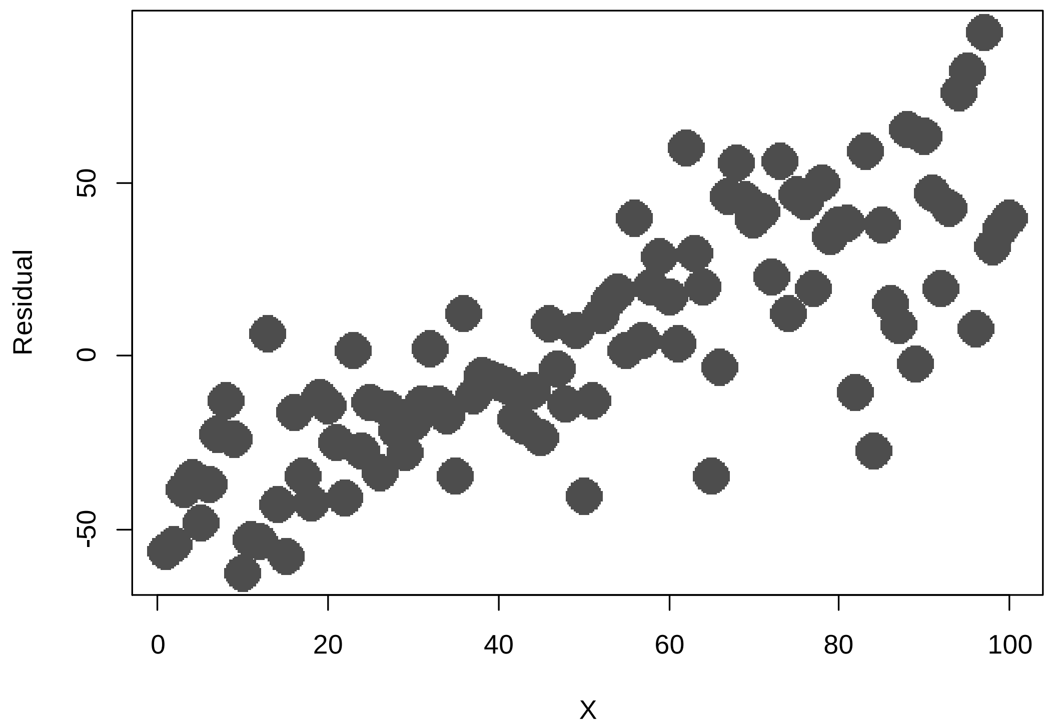


X = 1:100
Y = X*rnorm(100, 0, 1)

Fan-shaped

Residual plot for independence

Not Independent



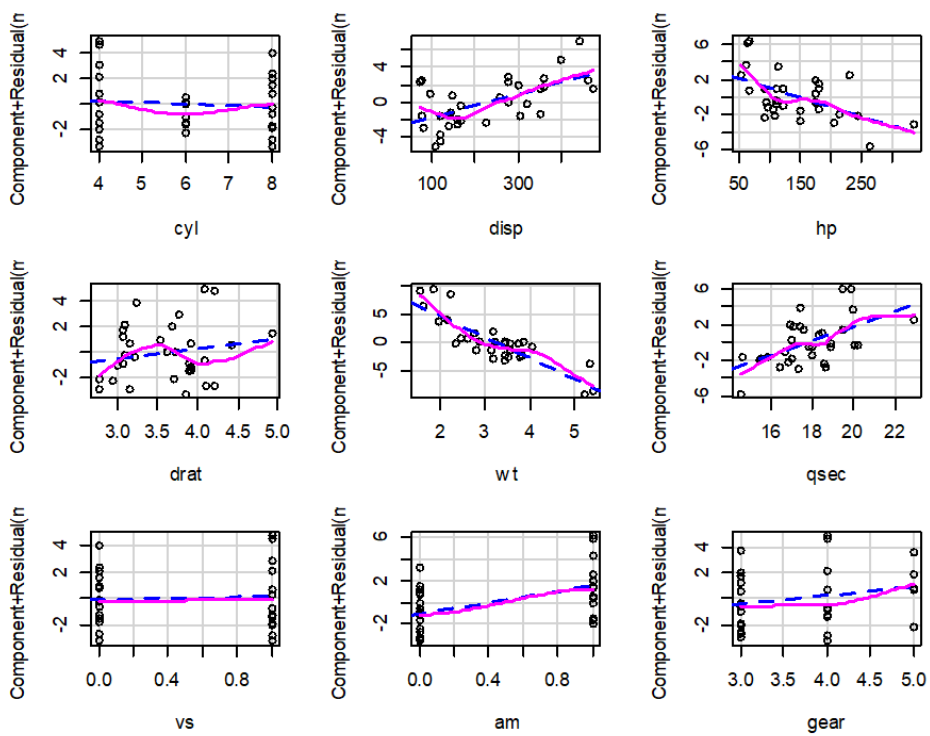
X = 1:100
Y = X+rnorm(100, 0, 20) - 50

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.

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

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:

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

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

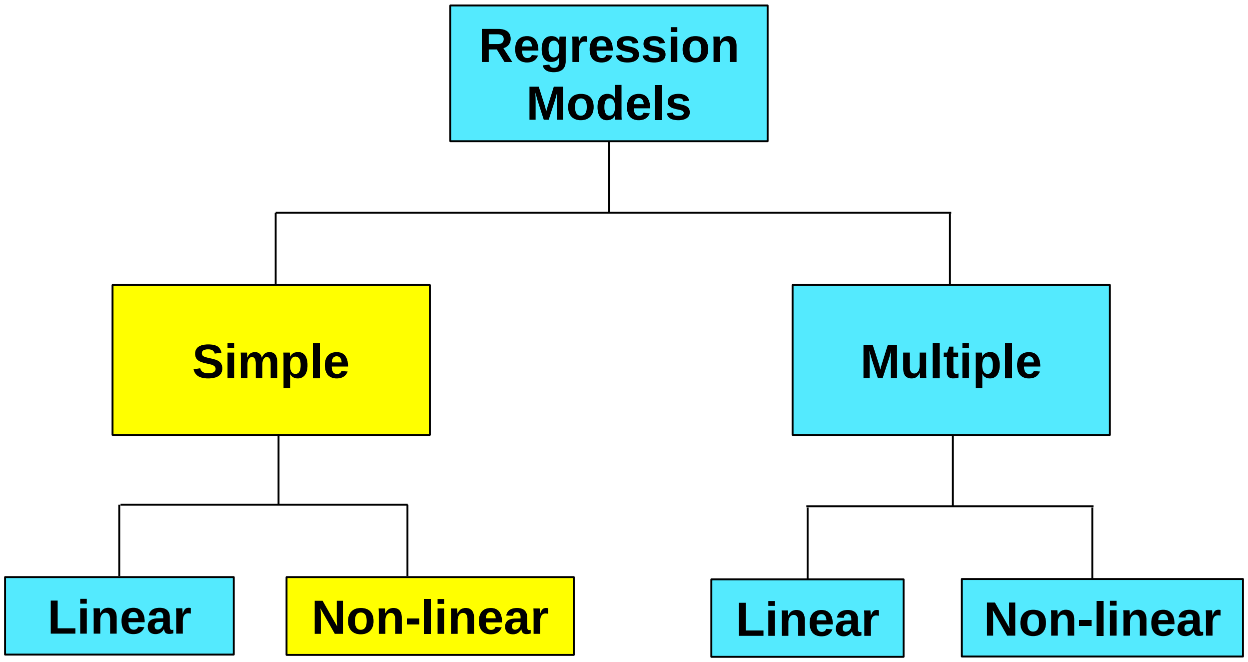
R code – VIF (variance inflation factor)

```
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

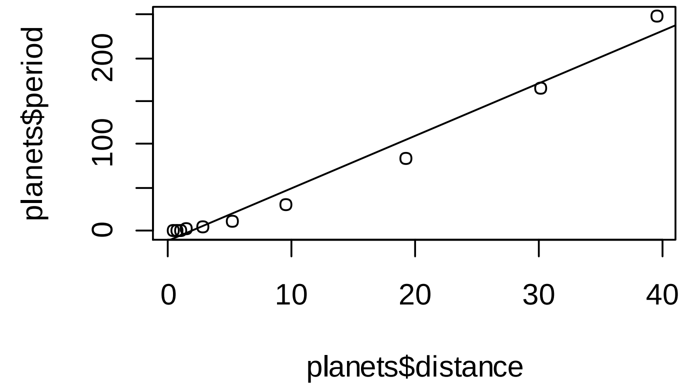
Types of Regression Models

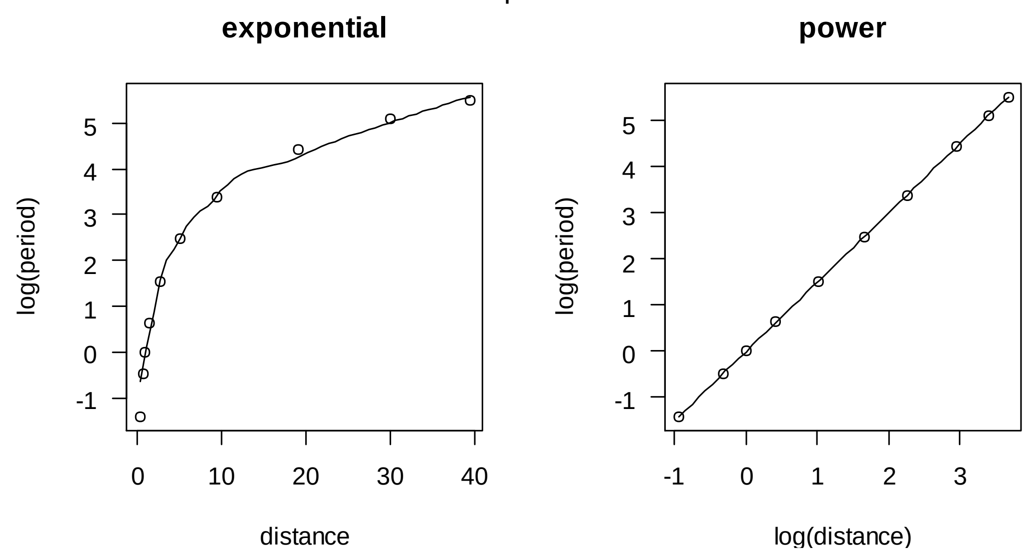


Power function

```
# 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))
```

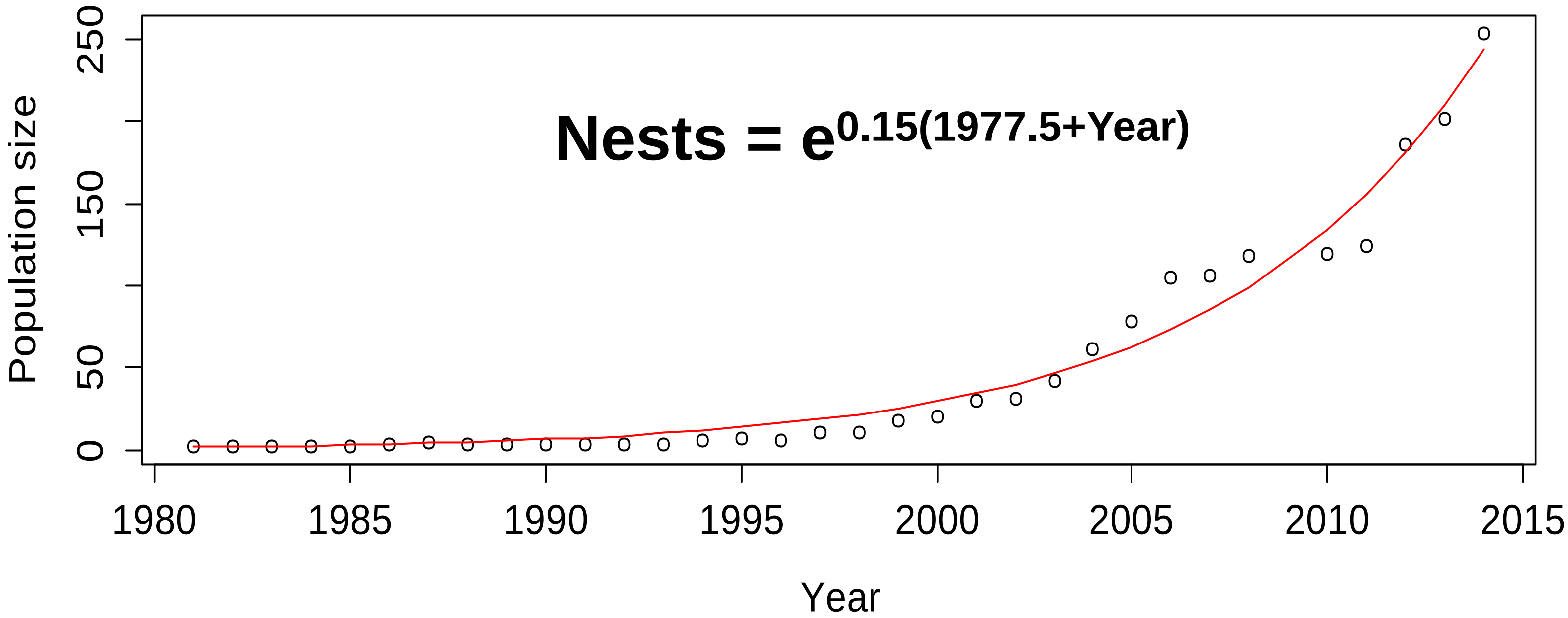
$period^2 = distance^3$






```
              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(Nest ~ 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

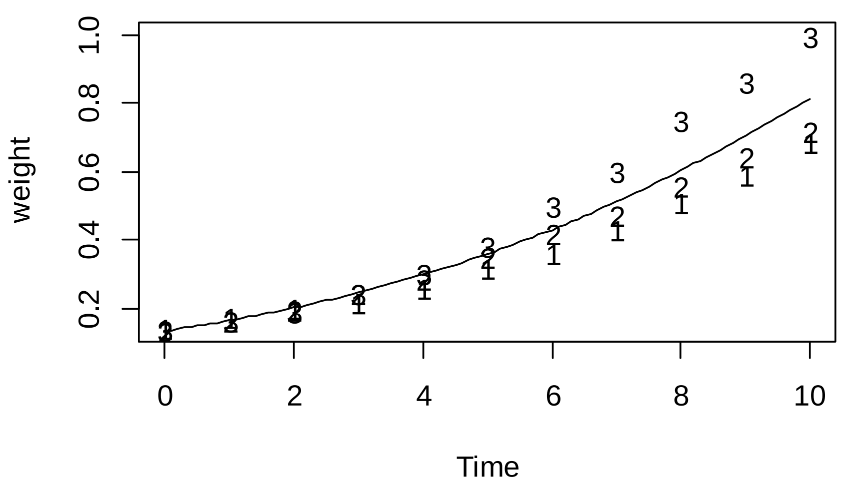
b0 = -1977.5; b1 = 0.15

residual sum-of-squares: 4279

Logistic function

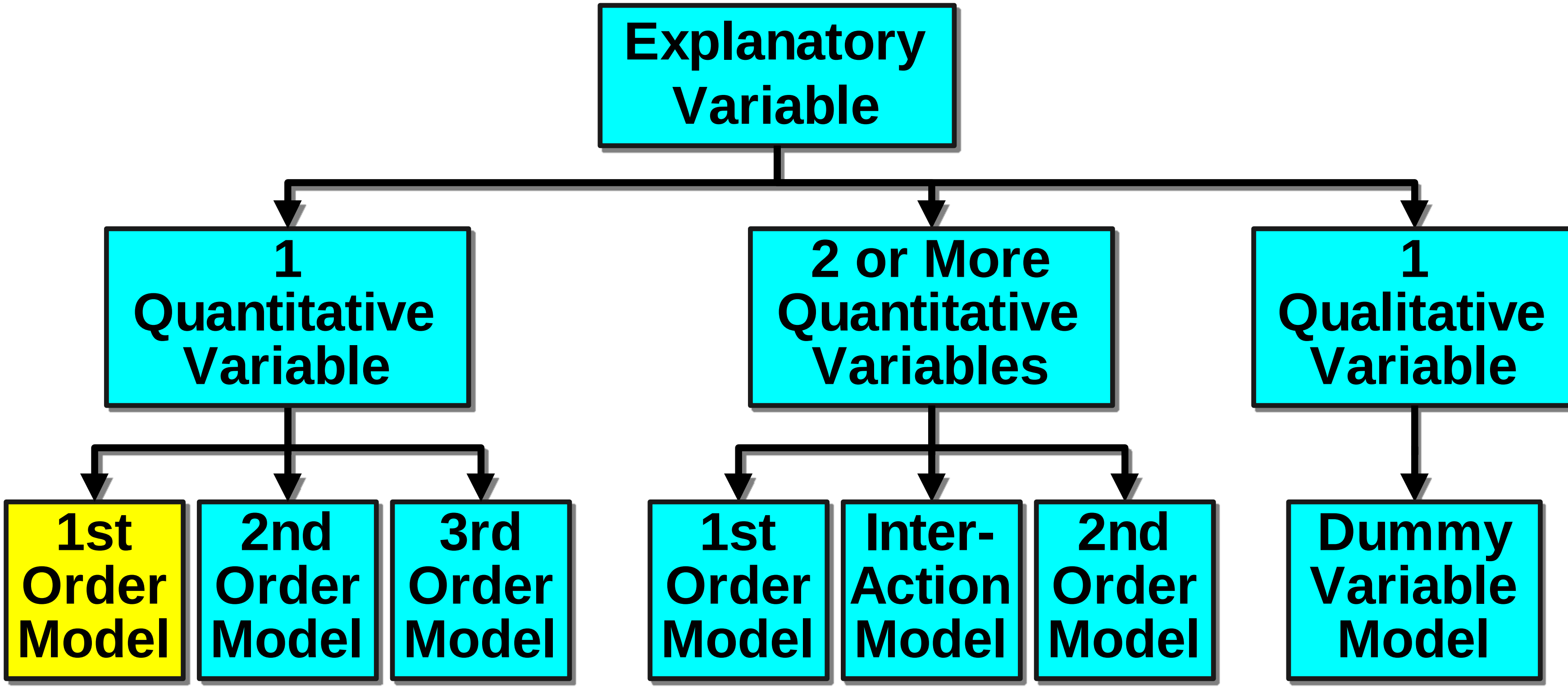
```
# Logistic growth
time <- c(seq(0,10),seq(0,10),seq(0,10))
plant <- c(rep(1,11),rep(2,11),rep(3,11))

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

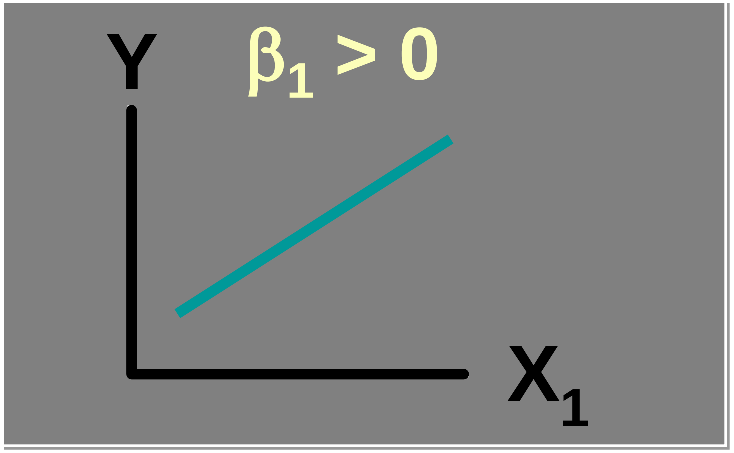
1. Relationship between 1 dependent & 1 independent variable is linear

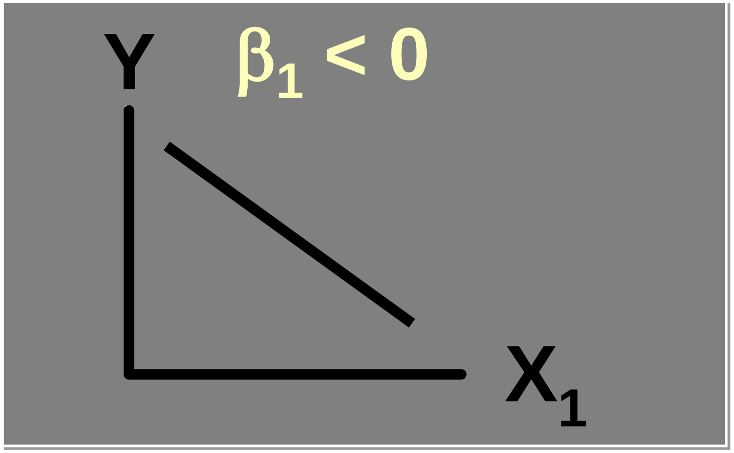
$$E(Y) = \beta_0 + \beta_1 X_{1i}$$

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

First-order model relationships

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



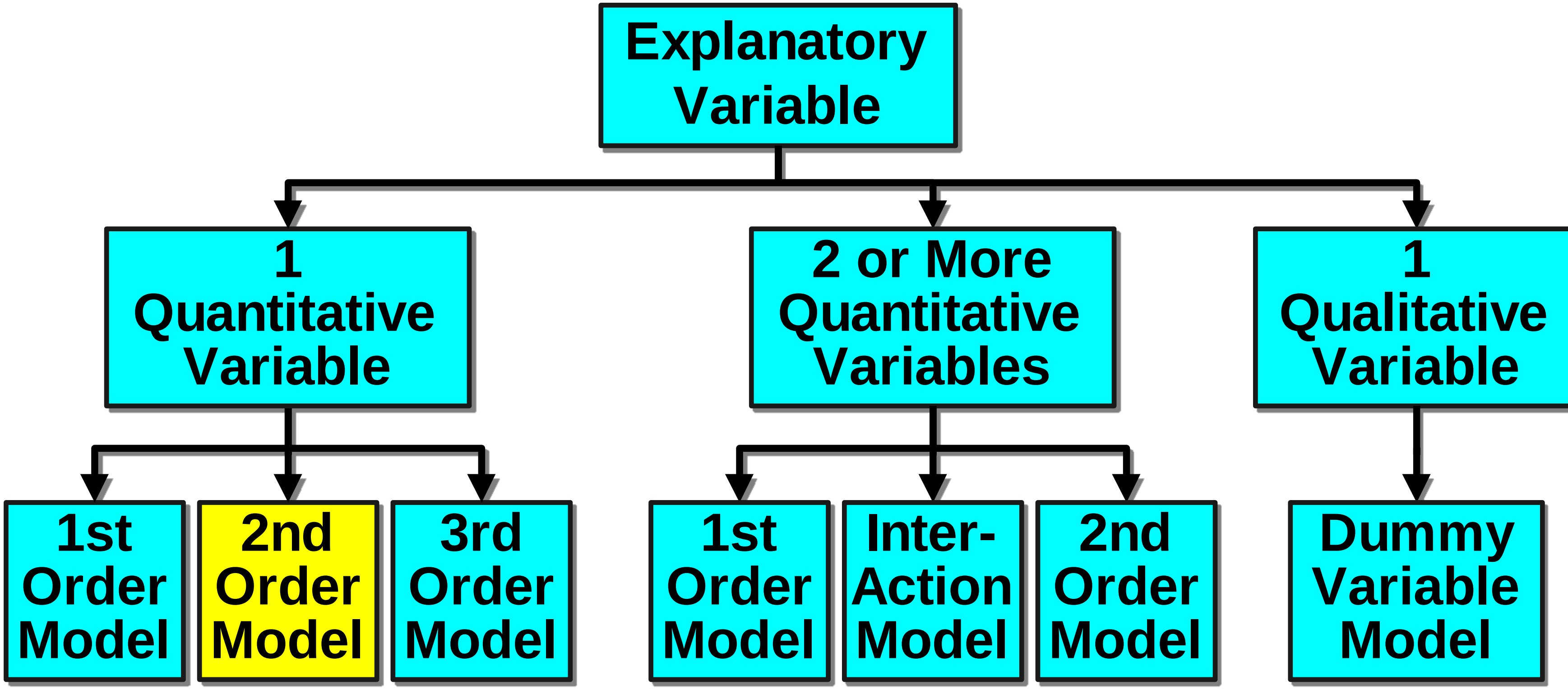


First-order model worksheet

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

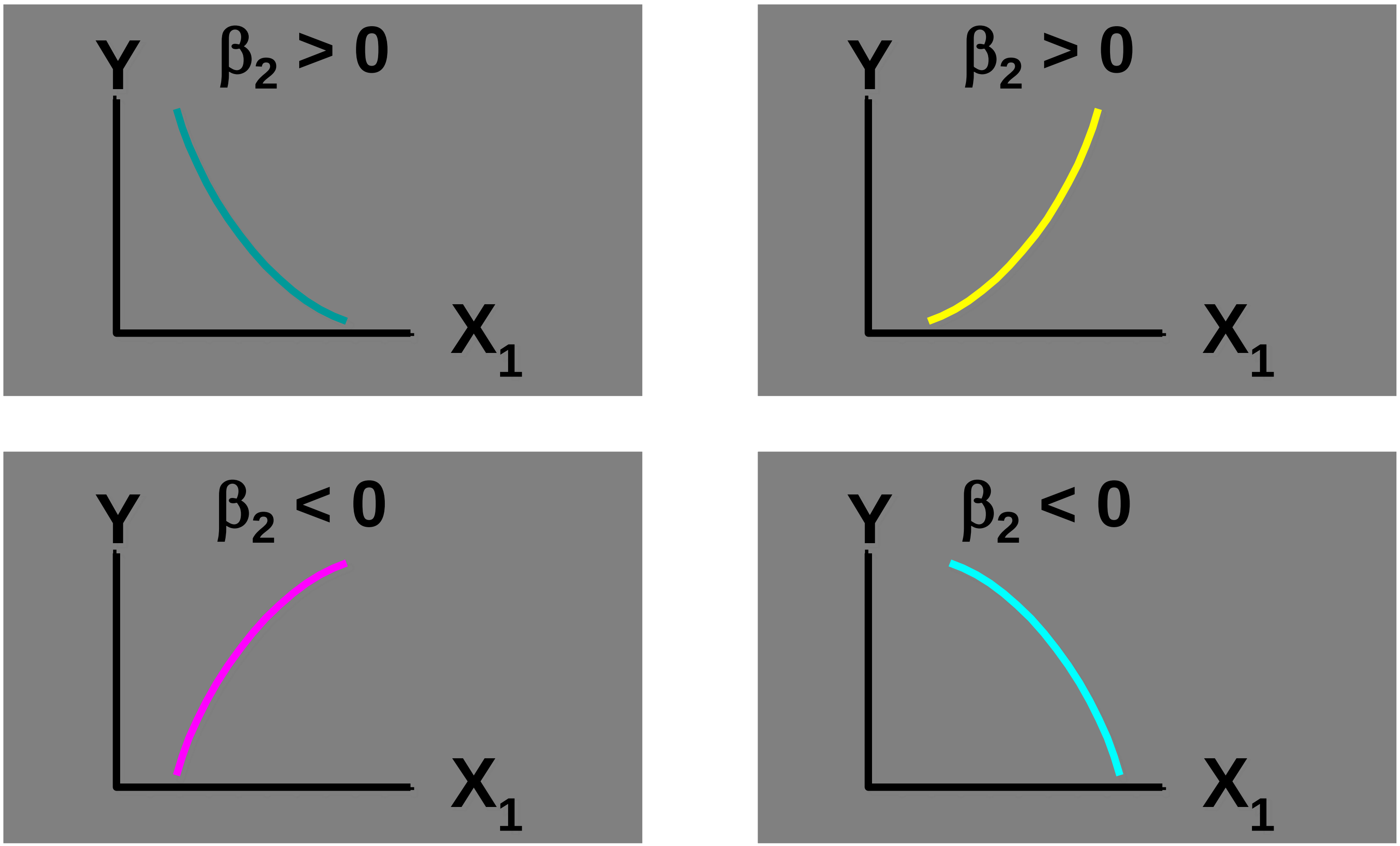
1. Relationship between 1 dependent & 1 independent variables is a quadratic function
2. Model:

Curvilinear effect

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

Linear effect

Second-order model relationships

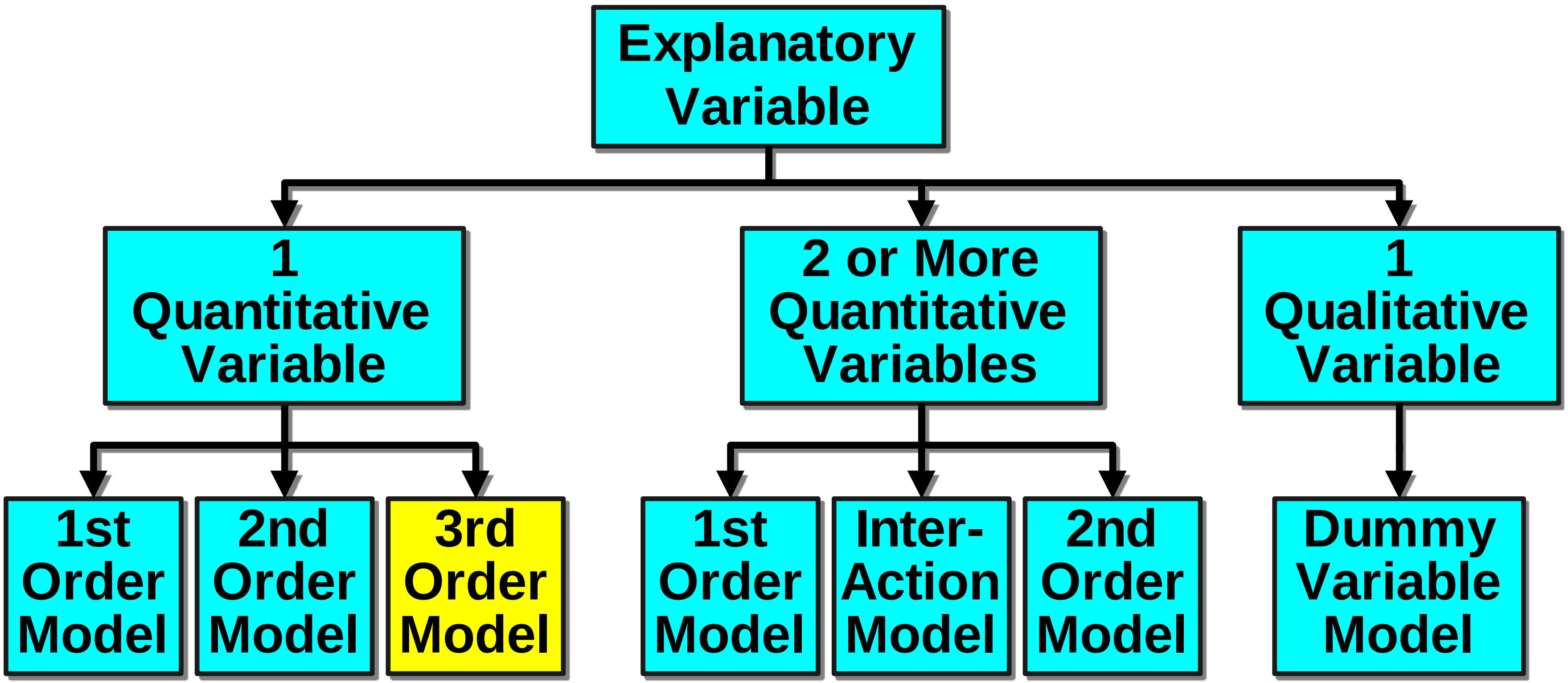


Second-order model worksheet

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:

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

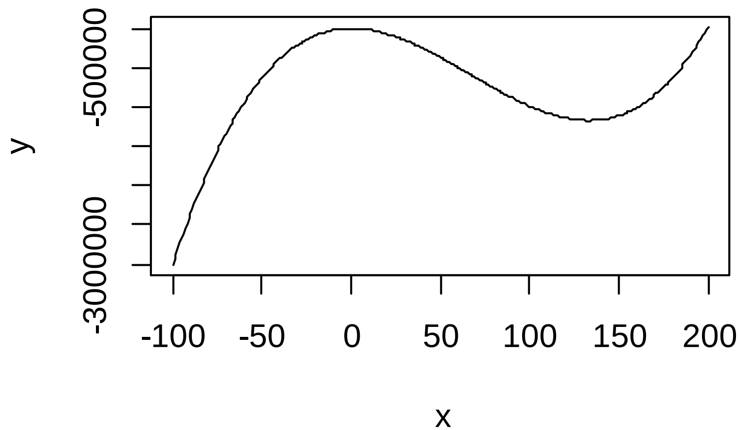
Linear effect

Curvilinear effects

Third-order model relationships

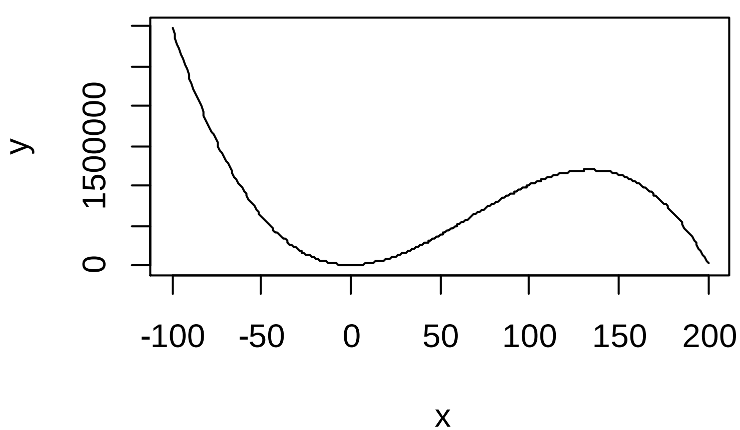
$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_1{}^2 + \beta_3 X_1{}^3$$

$\beta_3 > 0$



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

$\beta_3 < 0$



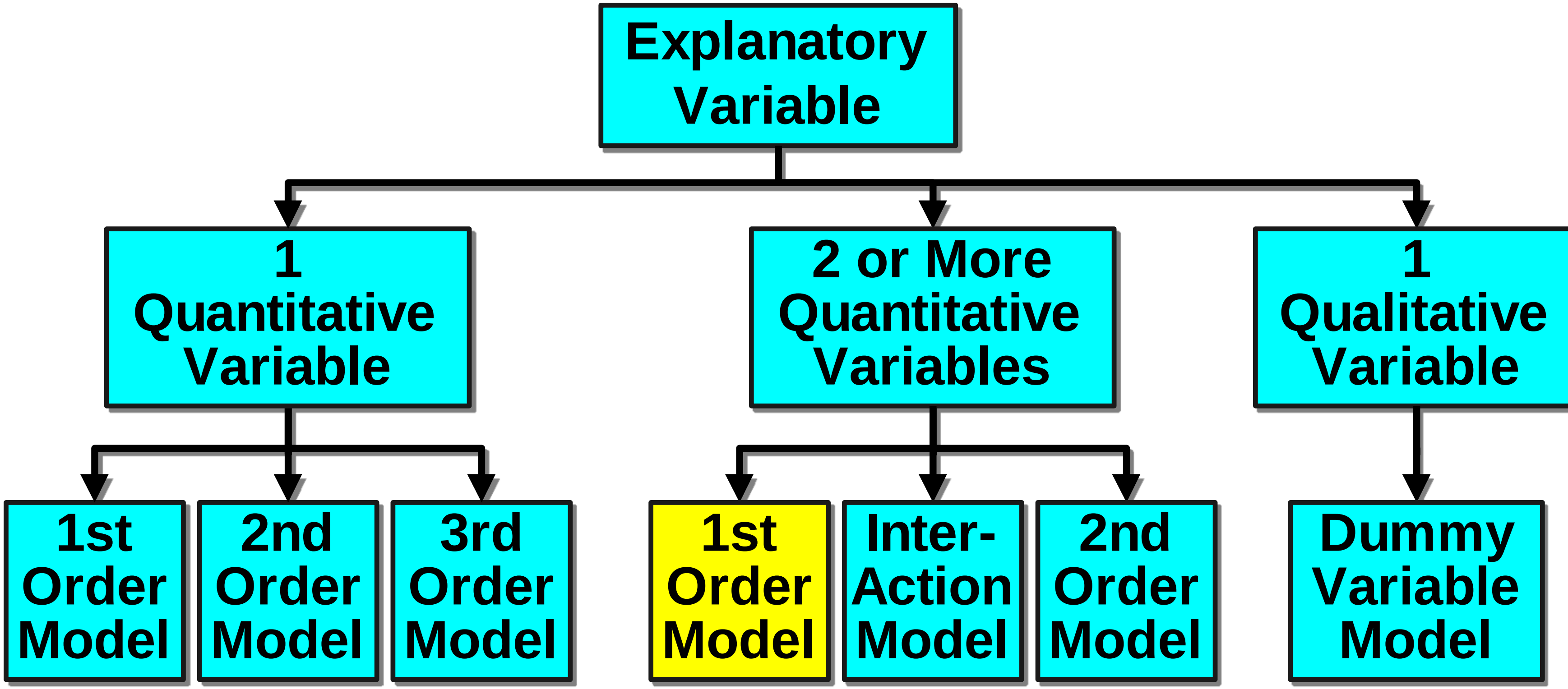
y = -x^3 + 200*x^2 + 100*x

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)

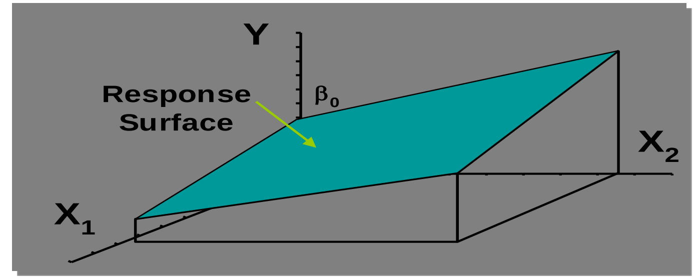


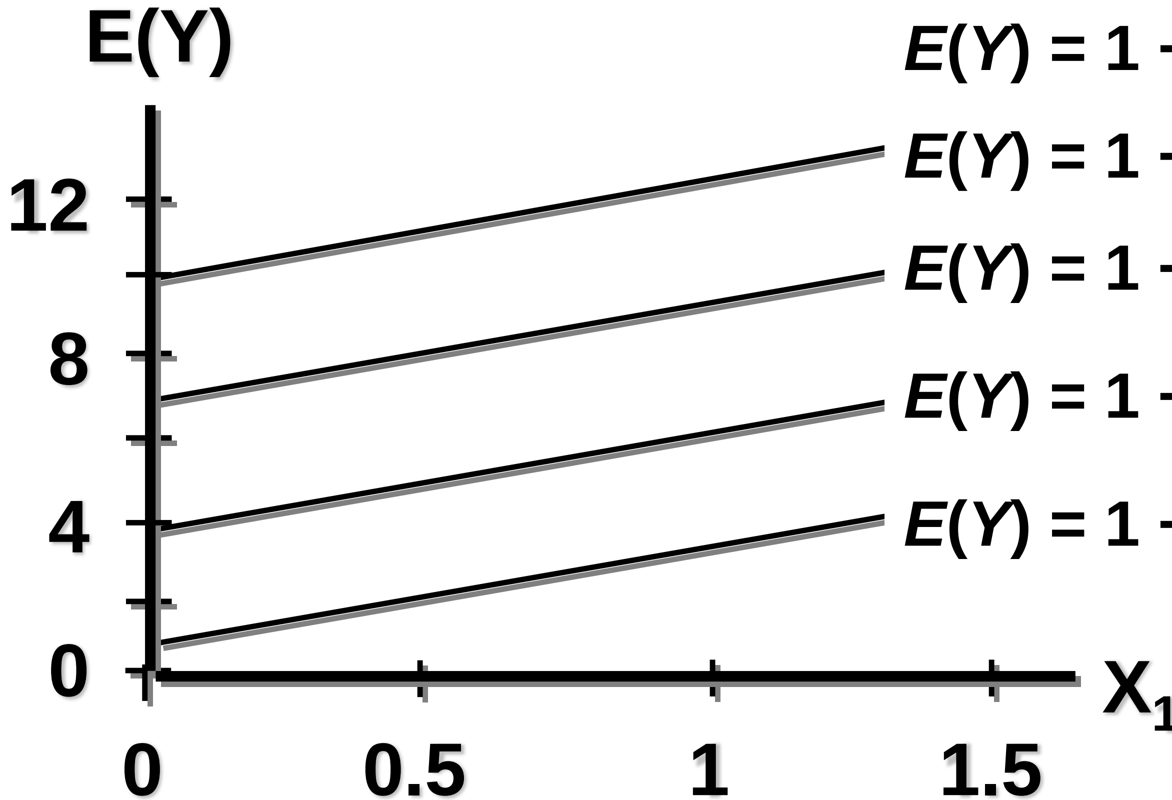
First-order model with 2 independent variables

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





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

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

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

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

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

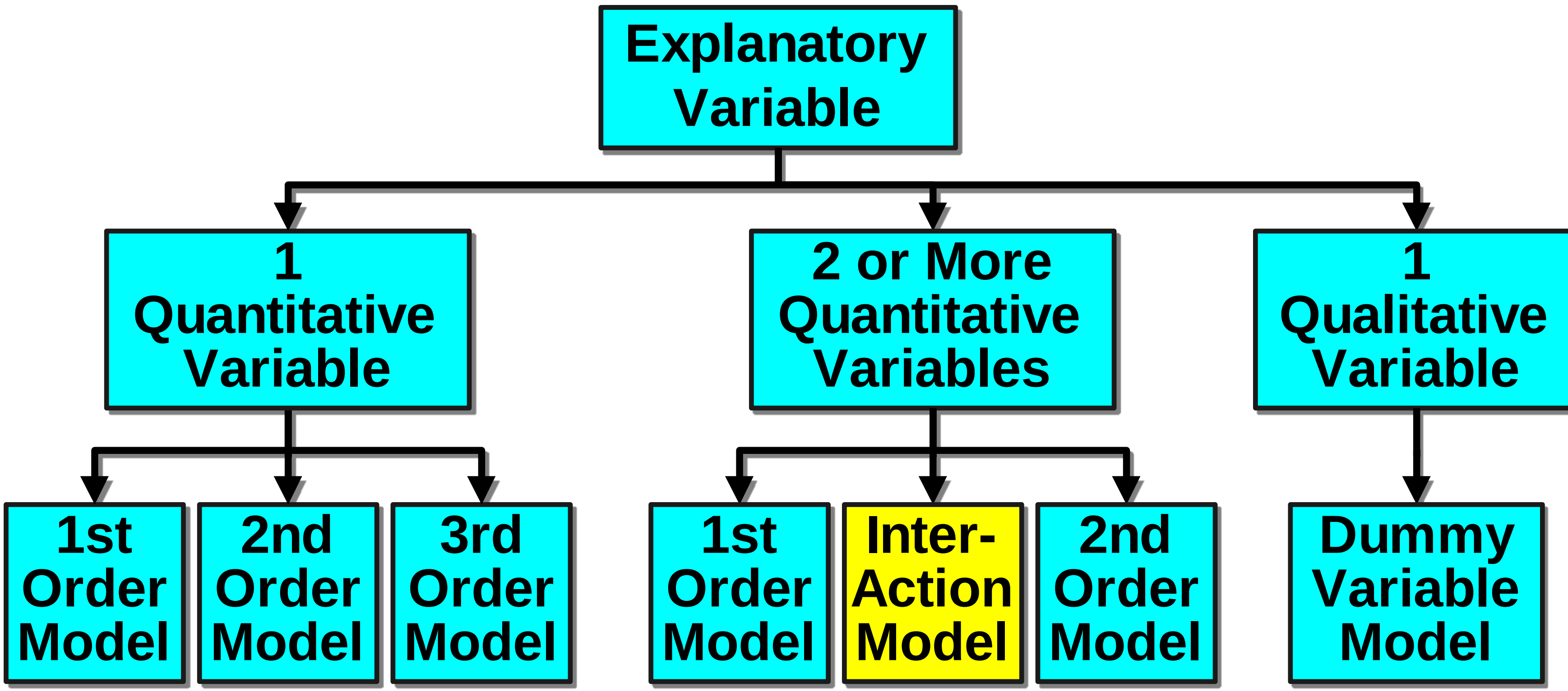
Effect (slope) of X_1 on $E(Y)$ does not depend on X_2 value

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

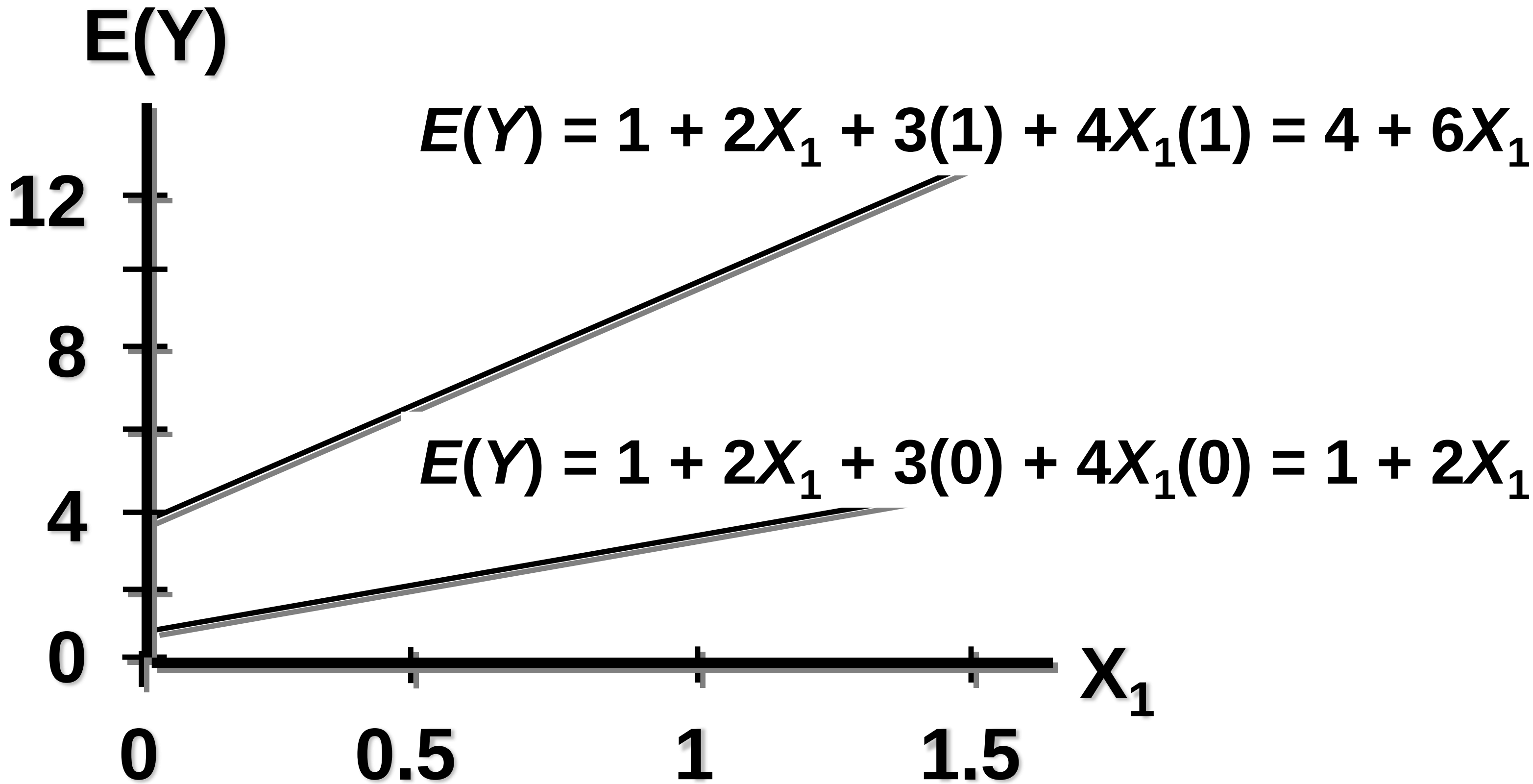
1. Given:

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

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

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



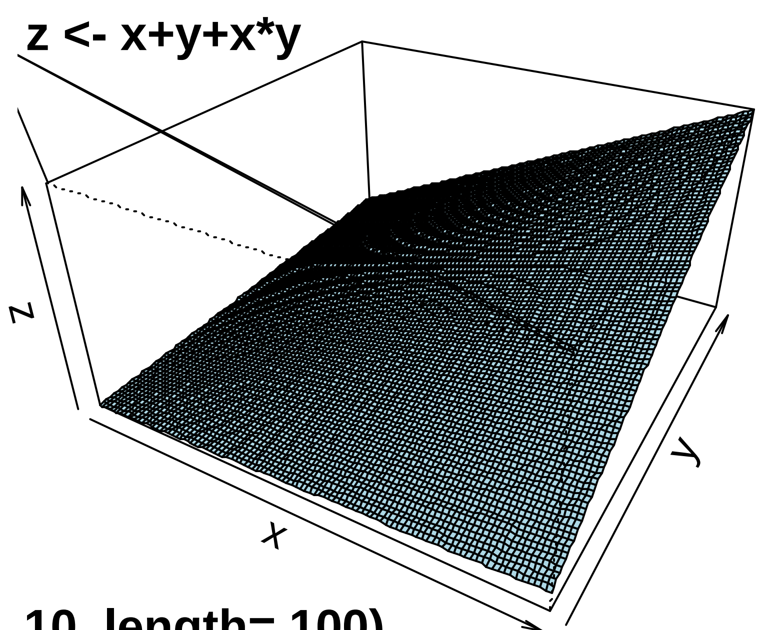
Effect (slope) of X_1 on $E(Y)$ does depend on X_2 value

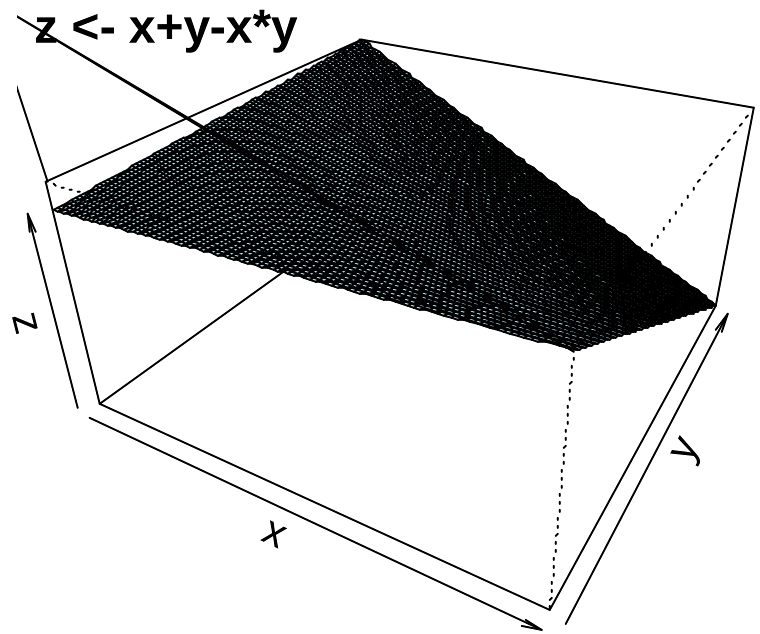
Interaction model worksheet

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

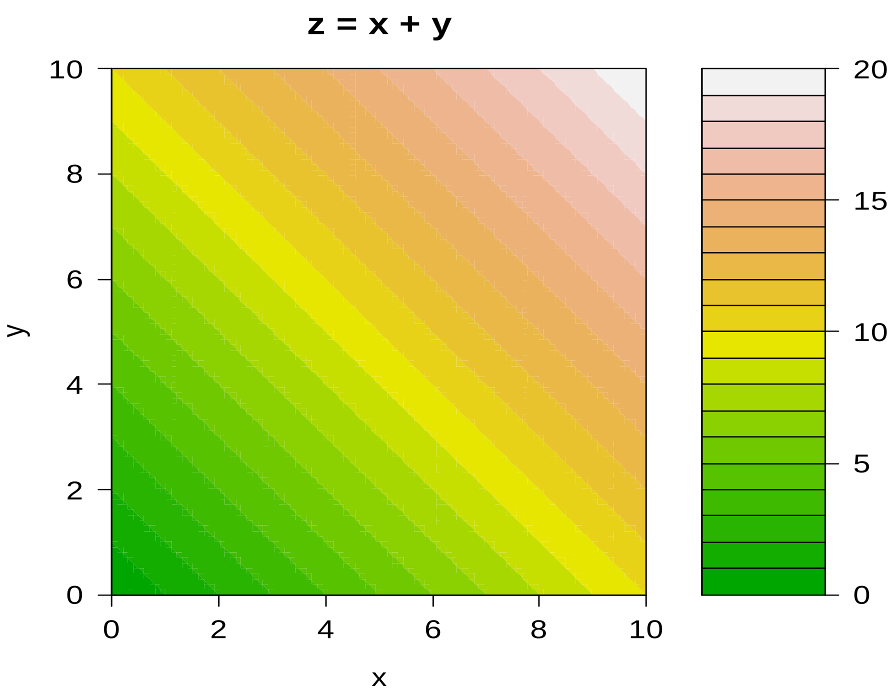


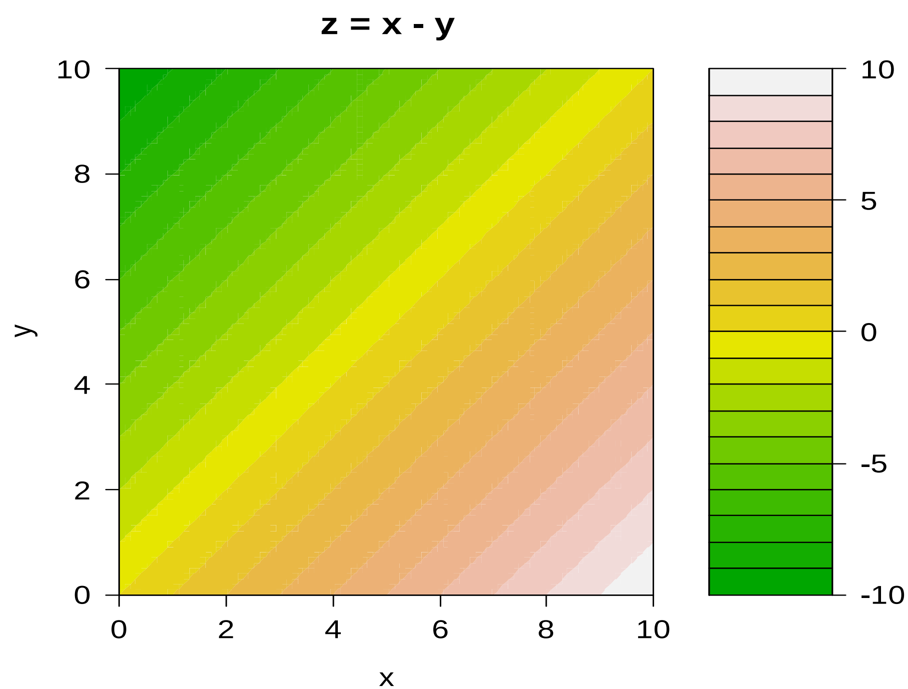



```
x <- seq(0, 10, length= 100)
y <- x
f <- function(x, y) { r <- x+y+x*y }
z <- outer(x, y, f)

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

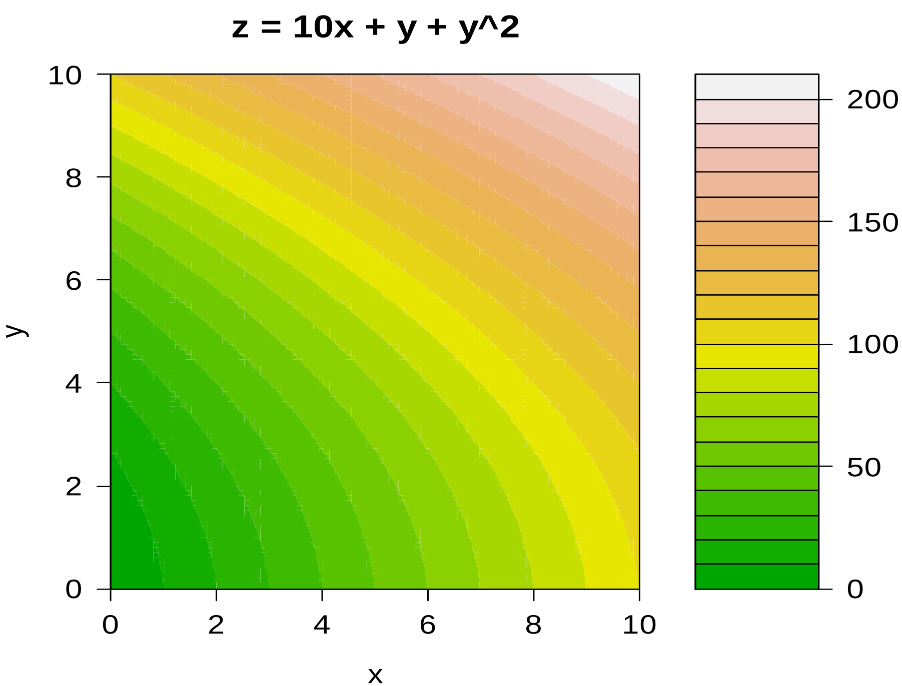
```
x = y <- seq(0, 10, length= 100); f <- function(x, y) { r <- x+y }; z <- outer(x, y, f)
    filled.contour(x, y, z, main="z = x + y", color = terrain.colors)
```

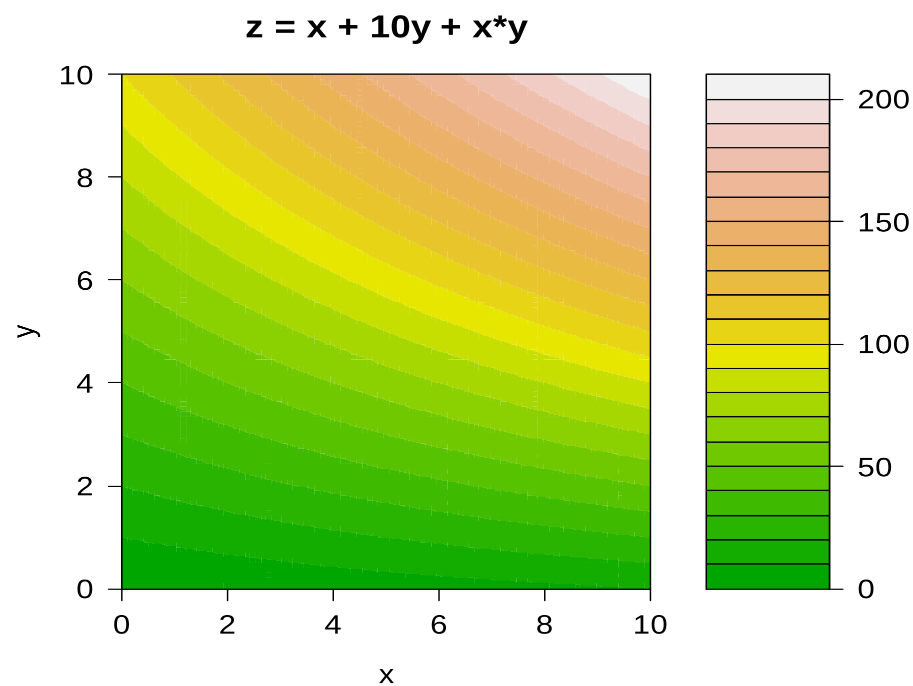




Contour plots for high order models

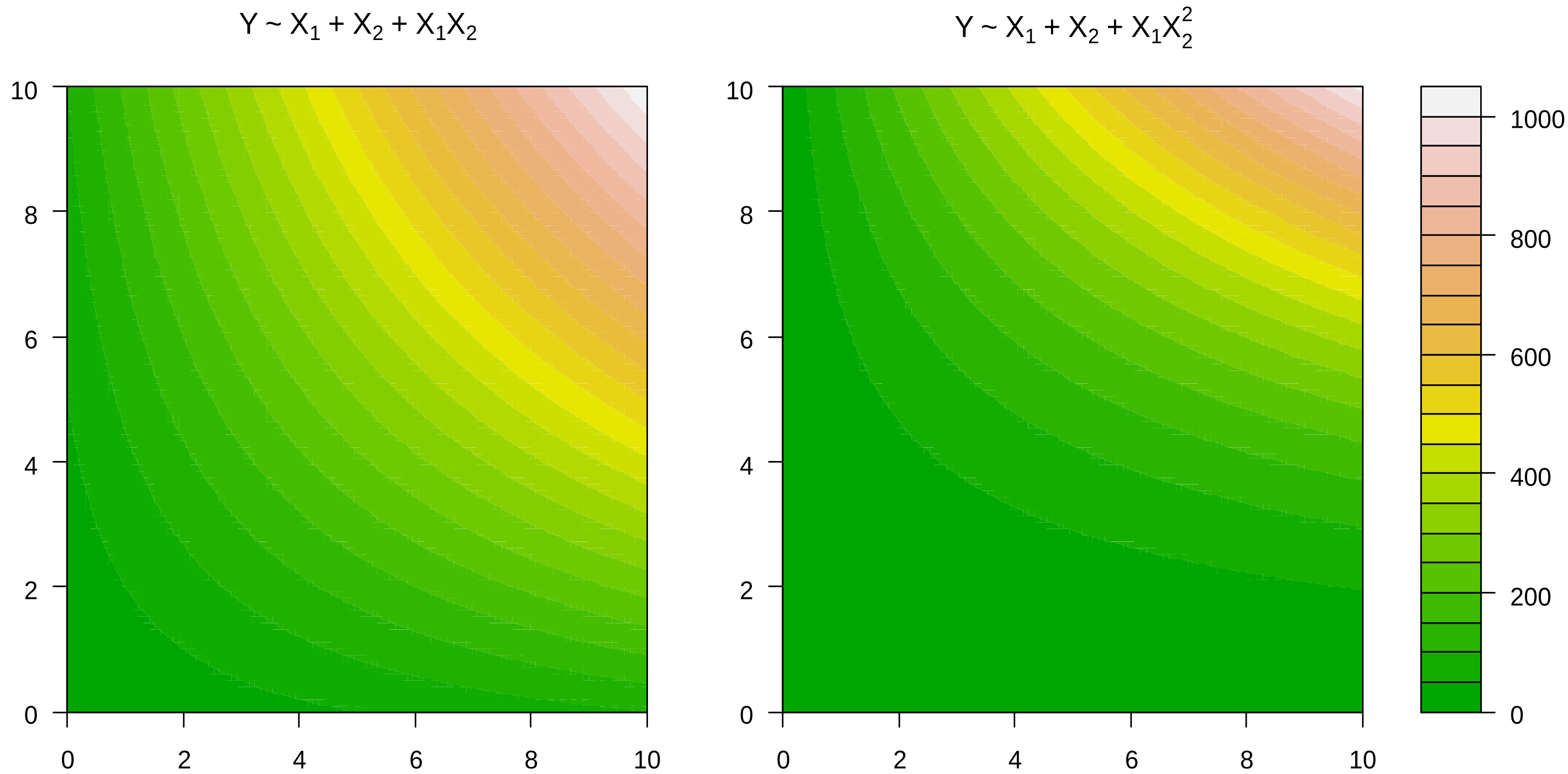
filled.contour(x, y, z, main="z = x + 10y + xy", color = terrain.colors)





Contour plots for interaction models

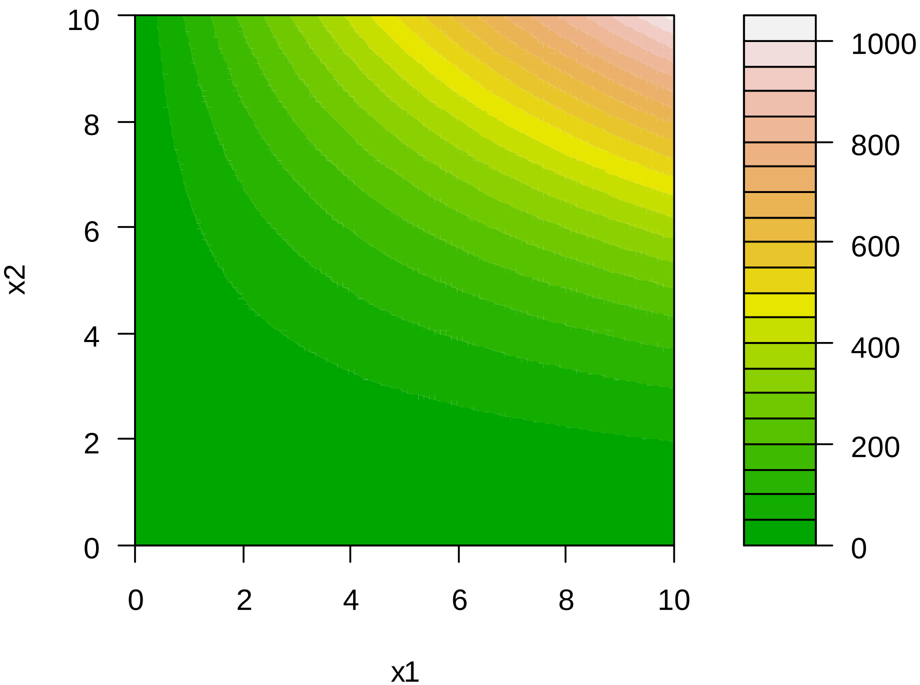
```
x1 = x2 <- seq(0, 10, length= 100); f <- function(x1, x2) { r <- x1+x2+x1*x2*x2 }; y <- outer(x1, x2, f)
filled.contour(x1, x2, y, main=expression(paste("Y ~ ", X[1], " + ", X[2], " + ", X[1], X[2]^2)), color = terrain.colors)
# [] subscript; ^ superscript
```



Estimating regression coefficients

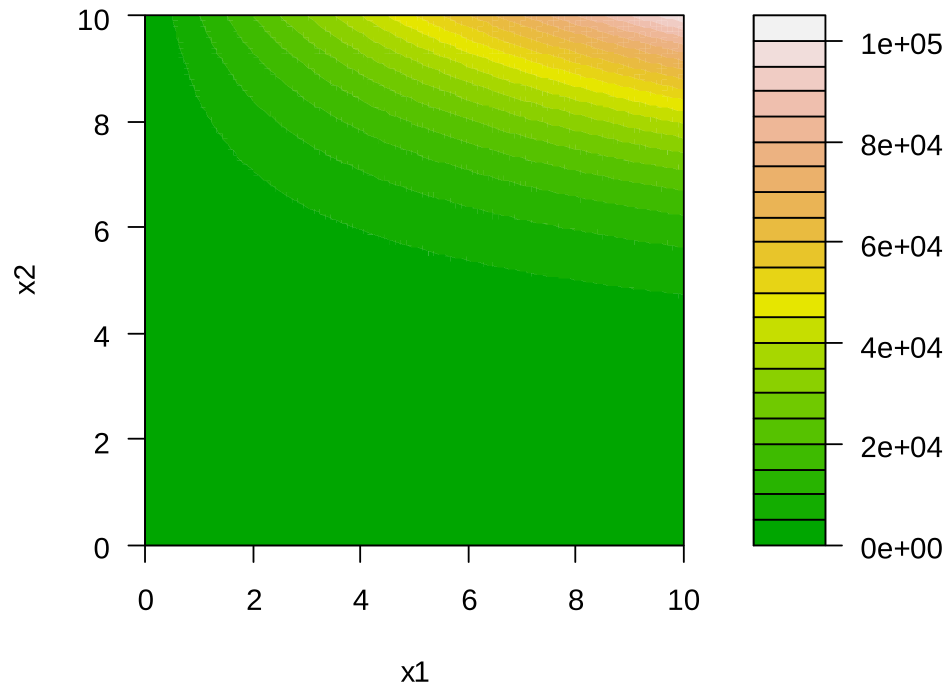
x1 = 0.8485
x2 = 1.0099
x1:x2 = 0.9787

y = x1 + x2 + x1*x2^2

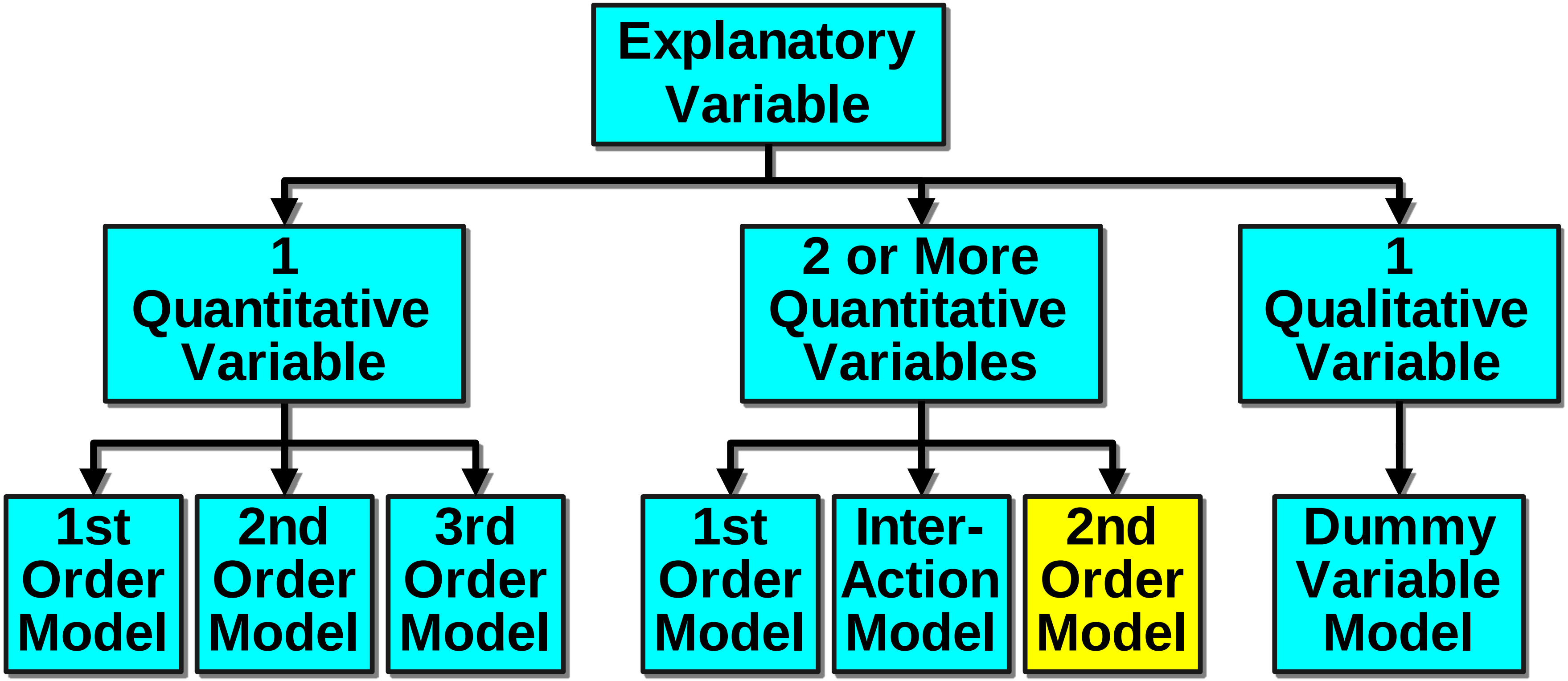


x1 = 0.8198
x2 = 1.0145
x1:x2 = 0.7744

y = x1 + x2 + x1*x2^4



Types of regression models (detailed)



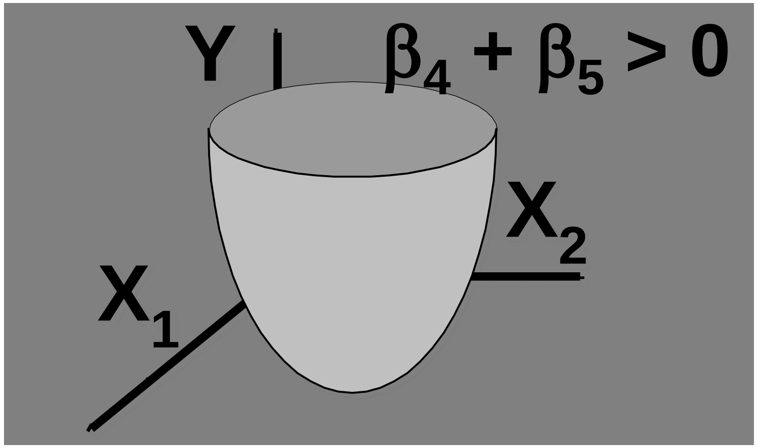
Second-order model with 2 independent variables

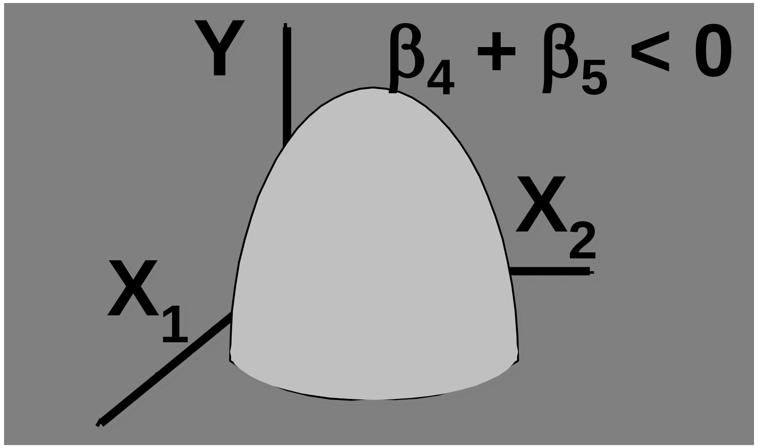
1. Relationship between 1 dependent & 2 or more independent variables is a quadratic function

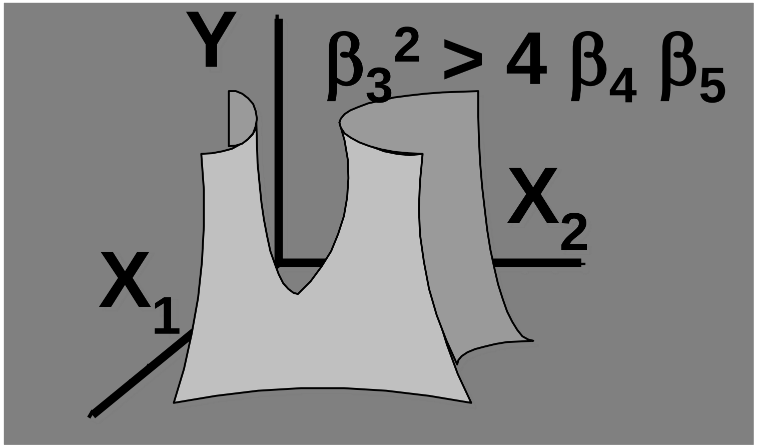
2. Use model:

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

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.

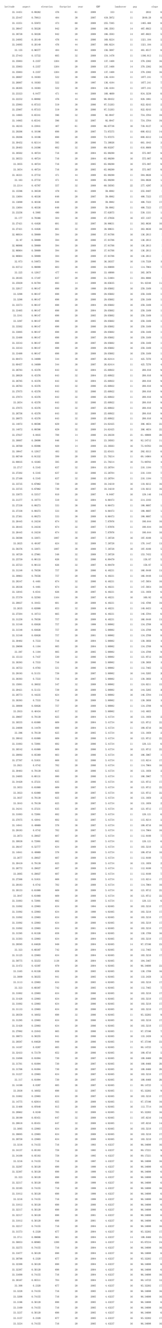
R code - multiple linear regression

```
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

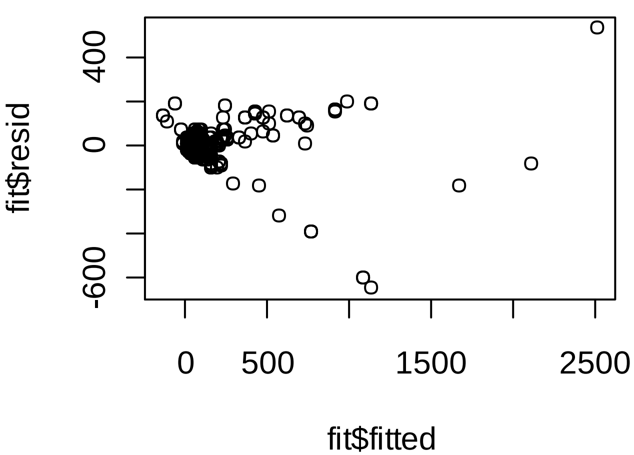
```
# Multiple Linear Regression Example (only include linear terms)
fit <- lm(pop ~ latitude+elevation+footprint+year+GDP+slope, data=ibis.pre)
summary(fit) # show results
```

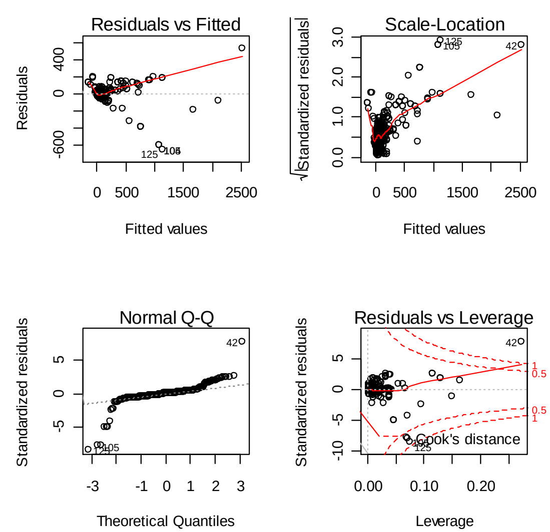
Coefficients:	Estimate	Std.Error	t value	Pr(>|t|)
(Intercept)	-8670.00000	2120.00000	-4.10000	0.00005
latitude	208.00000	49.80000	4.17000	0.00004
elevation	-0.14400	0.01930	-7.47000	0.00000
footprint	4.43000	0.62400	7.10000	0.00000
year	0.90300	0.64300	1.40000	0.16000
GDP	5.63000	0.11200	50.39000	<0.00000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



R code - multiple linear regression

```r
# 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)
```

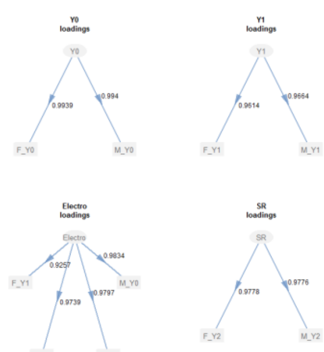
Path analysis

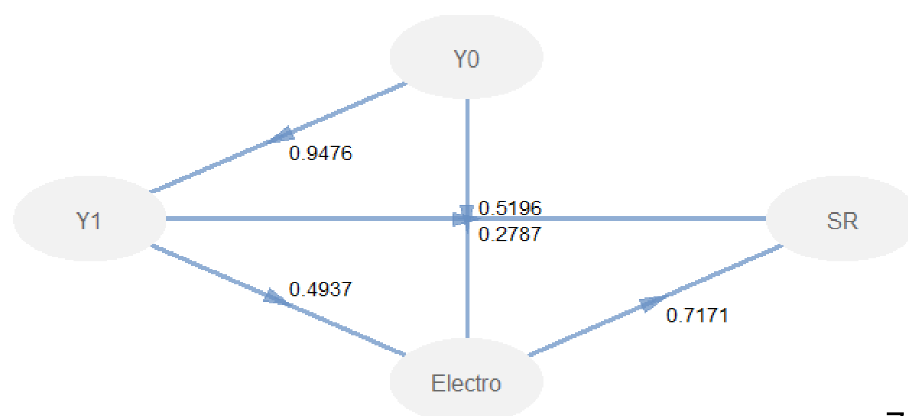
```
library(plspm)

D = read.csv('d:/data.csv', header=T);D
# path matrix (inner model realtionships)
Y0 = c(0, 0, 0, 0)
Y1 = c(1, 0, 0, 0)
Electro = c(1, 1, 0, 0)
SR = c(0,1,1,0)
saker_path = rbind(Y0, Y1, Electro, SR)
# add optional column names
colnames(saker_path) = rownames(saker_path)
# plot the path matrix
innerplot(saker_path)
# list indicating what variables are associated with what latent variables
saker_blocks = list(c(1,2), c(3,4),c(1,2,3,4),c(5,6))
# all latent variables are measured in a reflective way
saker_modes = rep("A", 4)
# run plspm analysis
saker_pls = plspm(D, saker_path, saker_blocks, modes = saker_modes)
# what's in saker_pls?
saker_pls
# path coefficients
saker_pls$path_coefs
# inner model
saker_pls$inner_model
# summarized results
summary(saker_pls)
# plot the results (inner model)
plot(saker_pls)
# plot the loadings of the outer model
plot(saker_pls, what = "loadings", arr.width = 0.2)
# plot the weights of the outer model
plot(saker_pls, what = "weights", arr.width = 0.1)
```

M_Y0	F_Y0	M_Y1	F_Y1	M_Y2	F_Y2
100	100	80	60	70	50
60	55	40	35	30	30
30	35	15	20	10	12
62	60	50	40	38	35
40	45	30	35	25	27
70	65	60	35	50	30
50	45	45	40	32	28
55	60	42	45	31	32

M_Y0 number of males at Year 0

	relationships	direct	indirect	total
1	Y0 -> Y1	0.948	0	0.948
2	Y0 -> Electro	0.52	0.468	0.987
3	Y0 -> SR	0	0.972	0.972
4	Y1 -> Electro	0.494	0	0.494
5	Y1 -> SR	0.279	0.354	0.633
6	Electro -> SR	0.717	0	0.717





Multiple correlation

Multiple correlation coefficient

- Correlation coefficient in the context of multiple regression
- *R* can be defined as the correlation between the criterion (*Y*) and the best linear combination of the predictors

$$R = r_{Y\hat{Y}}$$

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_p X_p$$

Partitioning of variance

- Sum of squares that is related to ***regression*** (*SSR*):

$$\sum(\hat{Y}_i - \bar{Y})^2 = SS_{\hat{Y}}$$

- Sum of squares that is related to ***residual*** (error; *SSE*):

$$\sum(Y_i - \hat{Y}_i)^2 = SS_{\mathrm{residual}}$$

- Sum of squares related to ***total*** deviation (*SST*):

$$\sum(Y_i - \bar{Y})^2 = SS_Y$$

Multiple correlation coefficient (squared)

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

- R^2 can be interpreted in terms of percentage of accountable variation
- $R^2 = 0.755$: we can say that 75.5% of the variation in *Y* can be predicted on the basis of the *X*s

Partial regression coefficients

- Coefficients in multiple regression are called ***partial regression coefficients***

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \cdots + b_p X_p$$

- Example: b_1 is coefficient for regression of Y on X_1 when we ***partial out*** the effect of X_2, …, X_p
  - When other variables are held constant
- Common mistake: equate b_1 ***in the context of*** the other X_i with the simple regression coefficient when ***ignoring*** X_i.

Partial correlation

Partial correlation measures the degree of association between two random variables, with the effect of a set of controlling random variables (e.g., x_2) removed for both variables (e.g., y and x_1)

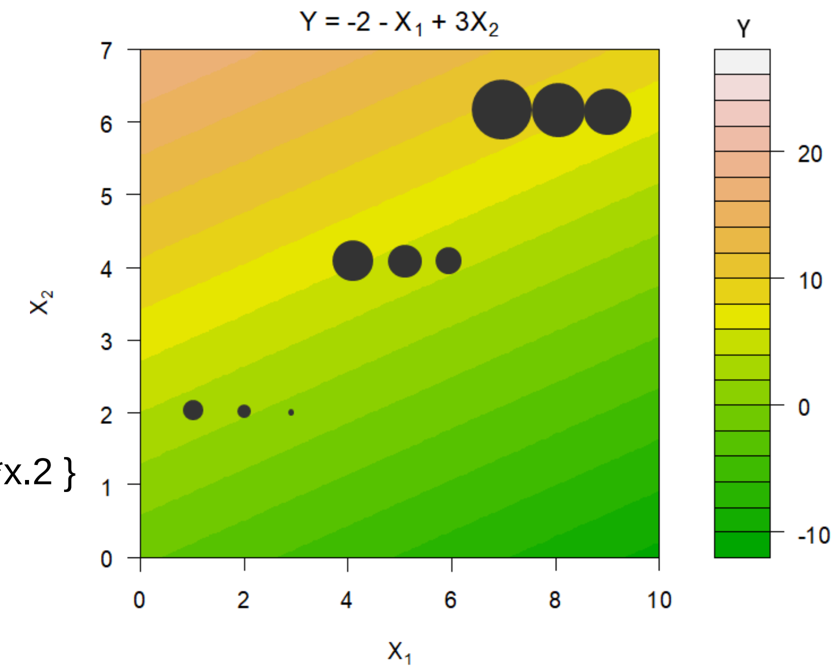
$y = x_1 + x_2$

$$Partial_{y1} = \frac{r_{y1} - (r_{y2})(r_{12})}{\sqrt{1 - r_{y2}^2}\sqrt{1 - (r_{12})^2}}$$

$$Semi - Partial_{y1} = \frac{r_{y1} - (r_{y2})(r_{12})}{\sqrt{1 - (r_{12})^2}}$$

Partial correlation example

```
y   <- c(3, 2, 1, 6, 5, 4, 9, 8, 7)
x1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
x2 <- c(2, 2, 2, 4, 4, 4, 6, 6, 6)
# multiple regression
fit = summary(lm(y ~ x1 + x2))
interception  = fit[[4]][1,1]
coef_x1       = fit[[4]][2,1]
coef_x2       = fit[[4]][3,1]
x.1 <- seq(min(x1) -1, max(x1) + 1, length= 100)
x.2 <- seq(min(x2) -2, max(x2) + 1, length= 100)
f <- function(x.1, x.2) { r <- interception + coef_x1*x.1 + coef_x2*x.2 }
y.pred <- outer(x.1, x.2, f)
filled.contour(x.1, x.2, y.pred, main="", color = terrain.colors,
               xlab=expression(paste(X[1])),
               ylab=expression(paste(X[2])),
               ylim=c(1, 7))
points(x1/1.3, x2, pch=16, cex=y)

library(ggm)# partial correlation
D = cbind(y, x1, x2)
D = jitter(D, factor = .01)
pcor(c("y", "x1"), var(D)) # 0.8
pcor(c("y", "x1","x2"), var(D)) # -0.9999
```



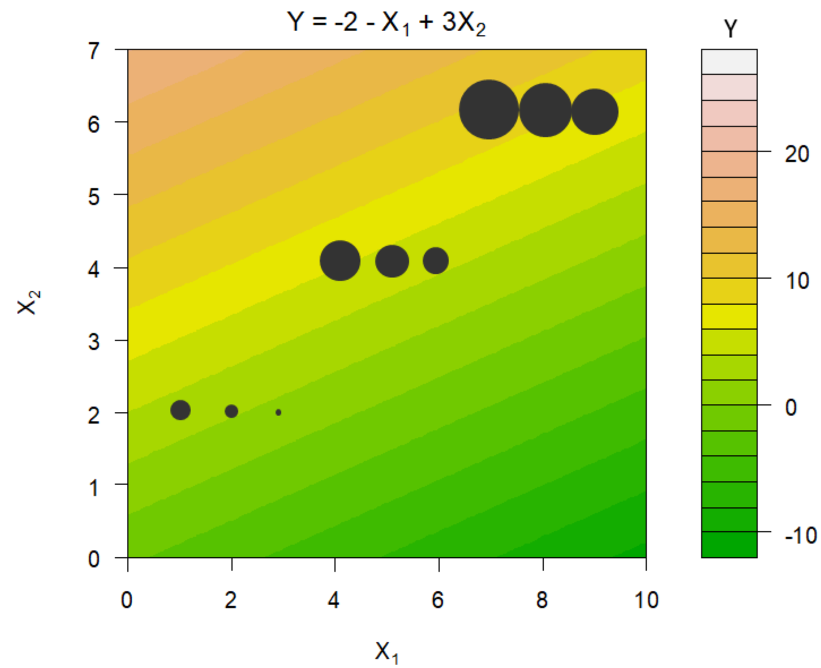
R code - partial correlation

```
# partial correlation
library(ggm)

## The marginal correlation between analysis and statistics
pcor(c("footprint", "GDP"), var(ibis.pre))
cor(ibis.pre$footprint, ibis.pre$GDP)
0.528

## The correlation between footprint and GDP given elevation
pcor(c("footprint", "GDP","elevation"), var(ibis.pre))
0.507

## The correlation between footprint and GDP given elevation and latitude
pcor(c("footprint", "GDP","elevation",“latitude"), var(ibis.pre))
0.5
```

Partial correlation example

- For any ***fixed*** value of X_2, slope of the regression line of Y on X_1 is negative (in fact $b_{01.2} = –0.99$)
- However, regression of Y on X_1 when ***ignoring*** X_2 is positive ($b_{01} = 0.8$)

Partialling out and ignoring are very different!



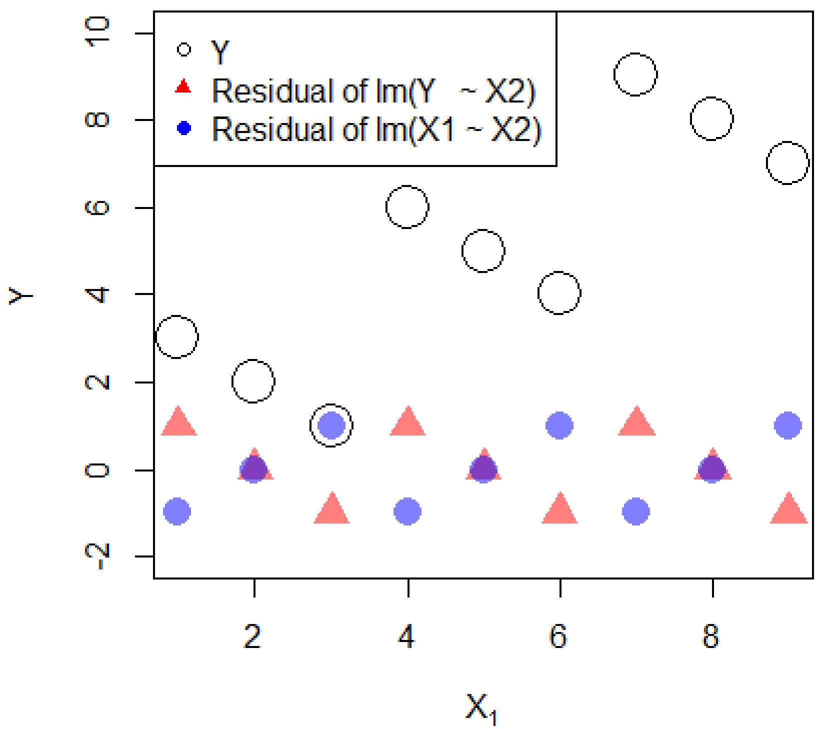
- When X_1 and X_2 are ***independent***, regression coefficients will be equal in both cases

Remove the effect of X_2

- Suppose we regress Y on X_2 and obtain the residual values $Y_r = Y_i - \hat{Y}_i$
- Residual values represent part of Y that cannot be predicted by X_2: ***independent*** of X_2
- Now regress X_1 on X_2 generating $X_{1r} = X_{1i} - \hat{X}_{1i}$
- Again, residual values represent part of X_1 that is ***independent*** of X_2
- We now have two sets of residuals: part of Y and part of X_1 that are ***independent*** of X_2
  – Partialled X_2 out of Y and out of X_1

Remove the effect of X_2

- Now regress Y_r on X_{1r}: regression coefficient will be the partial coefficient *b*
- Correlation between Y_r and X_{1r} is the ***partial correlation*** of Y and X_1, with X_2 partialled out: $r_{01.2}$



$$b_{Y_r X_{1r}} = \frac{\mathrm{cov}_{Y_r X_{1r}}}{s^2_{X_{1r}}} = -1$$

```
plot(x1, y, ylim=c(-2,10), pch=1, cex=3,
     xlab=expression(paste(X[1])), ylab="Y")
points(x1, resid.y, col=adjustcolor( "red", alpha.f = 0.5),
       pch=17, cex=2)
points(x1, resid.x1, col=adjustcolor( "blue", alpha.f = 0.5),
       pch=16, cex=2)
legend("topleft",
       c("Y", "Residual of lm(Y  ~ X2)", "Residual of lm(X1 ~ X2)"),
       pch=c(1, 17, 16), col=c(1,2,4))
```

Contribution, fraction, partial R^2

- Contribution of a variable x_j to the explanation of the variation of a dependent variable y.
- Fraction [a] in variation partitioning.
- Partial R^2 (partial determination coefficient) between an x_j and a y variable.

Contribution

Scherrer (1984) called the quantity $a_j * r_{yxj}$ the "contribution" of the *j*-th variable to the explanation of the variance of *y;*

– a_j is the standardized regression coefficient of the *j*-th explanatory variable,

– r_{yxj} is the simple correlation coefficient (Pearson *r*) between *y* and x_j.

Fraction [a] in variation partitioning

semipartial correlation squared

- This fraction measures the proportion of the variance of y explained by the explanatory variable x_1 (for example) when the other explanatory variables (x_2, x_3...) are held constant **with respect to x_1 only** (and not with respect to y).

- Thus, one obtains fraction [a] by examining the r^2 obtained by regressing y on the residuals of a regression of x_1 on x_2, x_3...

Partial R^2

- The **partial *R*** measures the mutual relationship between two variables *y* and x_j when other variables (x_1, x_2, x_3...) are held constant **with respect to the two variables involved y and x_j**

- The **partial R^2** is the square of the partial *R* above. For y = x_1+x_2, it measures the proportion of the variance of the residuals of *y* with respect to x_2 that is explained by the residuals of x_1 with respect to x_2.

R code for contribution, fraction, partial R^2 and variance partitioning

```
mtcars; mtcars.st = scale(mtcars); apply(mtcars.st, 2, var)
mtcars.st = as.data.frame(mtcars.st)

fit = lm(mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb, data=mtcars.st)

# Contribution
Contribution = coef(fit) * cor(mtcars.st)[1,]
fit2 = step(fit)
Contribution = coef(fit2)[-1] * cor(mtcars.st)[1, c(6,7,9)]

# Fraction
f.wt = lm(wt ~ qsec + am, data=mtcars.st)
res.wt = resid(f.wt)
Fraction.a = summary(lm(mtcars.st$mpg ~ res.wt))$r.squared

# Partial.R2
f.mpg = lm(mpg ~ qsec + am, data=mtcars.st)
res.mpg = resid(f.mpg)
Partial.R2 = summary(lm(res.mpg ~ res.wt))$r.squared

# Variance partition
fit = lm(mpg ~ wt + qsec + am, data=mtcars.st)
anova(fit)[[2]] / sum(anova(fit)[[2]])
```

	wt	qsec	am
Contribution	0.5516	0.1521	0.1458
Fraction [a]	0.1628	0.0968	
Partial R^2	0.5199	0.1175	
Variance	0.7528	0.0735	0.0232

‘wt’ (car weight measured in tons)
‘qsec’ (the number of second a car takes to reach .25 miles)
‘am’ (transmission type)

Canonical correlation analysis

- In CCA, there can be multiple response variables.
- Canonical correlations are the maximum correlation between a linear combination of the responses and a linear combination of the predictor variables.

Given a linear combination of X variables:

$$F = f_1X_1 + f_2X_2 + ... + f_pX_p$$

and a linear combination of Y variables:

$$G = g_1Y_1 + g_2Y_2 + ... + g_qY_q$$

The **first canonical correlation** is:

Maximum correlation coefficient between F and G,

for all F and G

$F_1=\{f_{11},f_{12},\ldots,f_{1p}\}$ and $G_1=\{g_{11},g_{12},\ldots,g_{1q}\}$

are corresponding **canonical variates**

One example of CCA

```
# http://www.ats.ucla.edu/stat/r/dae/canonical.htm

require(ggplot2)
require(GGally)
require(CCA)

# Example 1. A researcher has collected data on three psychological variables, four academic variables (standardized test scores)
# and gender for 600 college freshman. She is interested in how the set of psychological variables relates to the academic
# variables and gender.  In particular, the researcher is interested in how many dimensions (canonical variables) are necessary to
# understand the association between the two sets of variables.

mm <- read.csv("http://www.ats.ucla.edu/stat/data/mmreg.csv")
colnames(mm) <- c("Control", "Concept", "Motivation", "Read", "Write", "Math",  "Science", "Sex")
summary(mm); head(mm)

psych <- mm[, 1:3]
acad <- mm[, 4:8]

ggpairs(psych)
ggpairs(acad)
```

Control	Concept	Motivation	Read	Write	Math	Science	Sex
-0.84	-0.24	1	54.8	64.5	44.5	52.6	1
-0.38	-0.47	0.67	62.7	43.7	44.7	52.6	1
0.89	0.59	0.67	60.6	56.7	70.5	58	0
0.71	0.28	0.67	62.7	56.7	54.7	58	0
-0.64	0.03	1	41.6	46.3	38.4	36.3	1
1.11	0.9	0.33	62.7	64.5	61.4	58	1

```
# correlations within and between the two sets of variables
matcor(psych, acad)  #CCA package
```

One example of CCA

```
# Canonical Correlation Analysis
cc1 <- cc(psych, acad)
summary(cc1)
```

```
##          Length   Class    Mode
## cor      3        -none-   numeric
## names    3        -none-   list
## xcoef    9        -none-   numeric
## ycoef    15       -none-   numeric
## scores   6        -none-   list
```

```
# display the canonical correlations
cc1$cor
```

```
## [1] 0.4641    0.1675    0.1040
```

```
# raw canonical coefficients
cc1[3:4]
```

```
## $xcoef
##              [,1]       [,2]       [,3]
## Control     -1.2538    -0.6215    -0.6617
## Concept      0.3513    -1.1877     0.8267
## Motivation  -1.2624     2.0273     2.0002
##
## $ycoef
##             [,1]        [,2]        [,3]
## Read       -0.044621   -0.004910    0.021381
## Write      -0.035877    0.042071    0.091307
## Math       -0.023417    0.004229    0.009398
## Science    -0.005025   -0.085162   -0.109835
## Sex        -0.632119    1.084642   -1.794647
```

```
# compute canonical loadings
cc2 <- comput(psych, acad, cc1)
```

```
# display canonical loadings
cc2[3:6]
```

```
## $corr.X.xscores
##               [,1]       [,2]      [,3]
## Control     -0.90405   -0.3897   -0.1756
## Concept     -0.02084   -0.7087    0.7052
## Motivation  -0.56715    0.3509    0.7451
##
## $corr.Y.xscores
##              [,1]       [,2]       [,3]
## Read        -0.3900   -0.06011    0.01408
## Write       -0.4068    0.01086    0.02647
## Math        -0.3545   -0.04991    0.01537
## Science     -0.3056   -0.11337   -0.02395
## Sex         -0.1690    0.12646   -0.05651
##
## $corr.X.yscores
##               [,1]        [,2]       [,3]
## Control     -0.419555   -0.06528   -0.01826
## Concept     -0.009673   -0.11872    0.07333
## Motivation  -0.263207    0.05878    0.07749
##
## $corr.Y.yscores
##             [,1]      [,2]       [,3]
## Read      -0.8404   -0.35883    0.1354
## Write     -0.8765    0.06484    0.2546
## Math      -0.7639   -0.29795    0.1478
## Science   -0.6584   -0.67680   -0.2304
## Sex       -0.3641    0.75493   -0.5434
```

http://www.ats.ucla.edu/stat/r/dae/canonical.htm

One example of CCA

In general, the number of canonical dimensions is equal to the number of variables in the smaller set, however, the number of significant dimensions may be even smaller.

Canonical dimensions, also known as canonical variates, are latent variables that are analogous to factors obtained in factor analysis.

For this particular model there are three canonical dimensions of which only the first two are statistically significant.

```
# tests of canonical dimensions
ev <- (1 - cc1$cor^2)

n <- dim(psych)[1]
p <- length(psych)
q <- length(acad)
k <- min(p, q)
m <- n - 3/2 - (p + q)/2
w <- rev(cumprod(rev(ev)))

# initialize
d1 <- d2 <- f <- vector("numeric", k)

for (i in 1:k) {
  s <- sqrt((p^2 * q^2 - 4)/(p^2 + q^2 - 5))
  si <- 1/s
  d1[i] <- p * q
  d2[i] <- m * s - p * q/2 + 1
  r <- (1 - w[i]^si)/w[i]^si
  f[i] <- r * d2[i]/d1[i]
  p <- p - 1
  q <- q - 1
}

pv <- pf(f, d1, d2, lower.tail = FALSE)
(dmat <- cbind(WilksL = w, F = f, df1 = d1, df2 = d2, p = pv))
```

http://www.ats.ucla.edu/stat/r/dae/canonical.htm

Assignment

General objectives: learn about multiple linear regression.

- Develop a dataset to perform:
  - Multiple linear regression Y-X1, X2, X3, etc.
- Check R^2, significance of each variables, multicollinearity, homogeneous of residuals
- Briefly interpret the results

R code – correlation plot

```
## put (absolute) correlations on the upper panels,
## with size proportional to the correlations.
panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1)) # ranges for x-axis and y-axis in plots
  r <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste(prefix, txt, sep = "")
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

pairs(ibis.pre, lower.panel = panel.smooth,
      upper.panel = panel.cor)
```