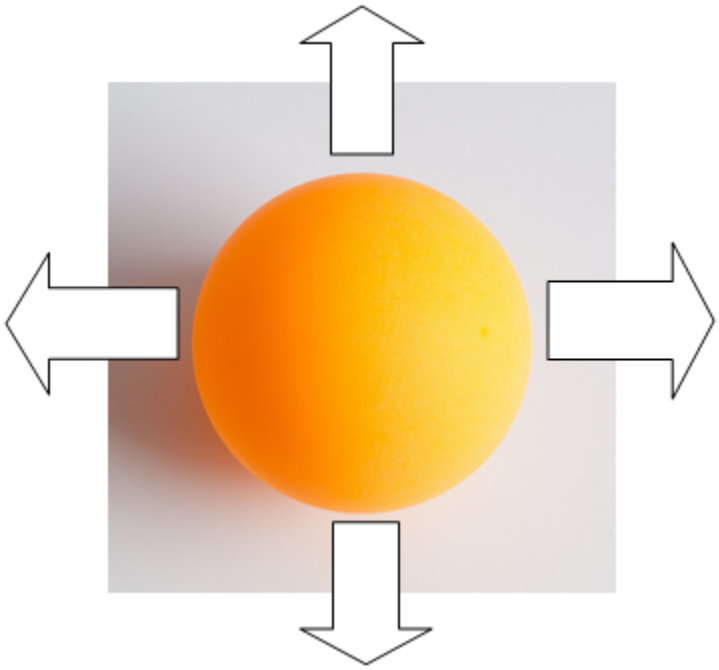
Appetizer

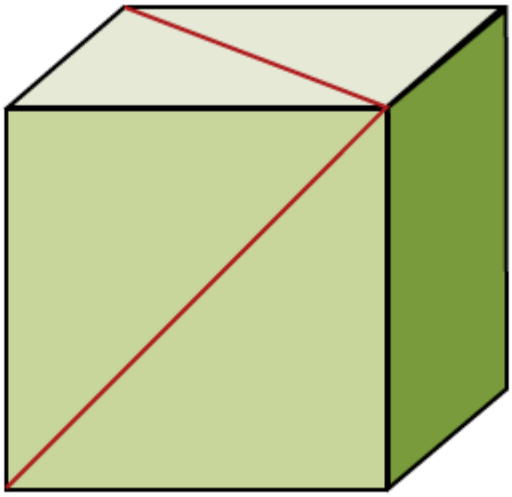
#1. Metals expand when heated up.



When a metal washer is heated up …

the "inside" hole will
(a) get bigger
(b) get smaller
(c) remain unchanged

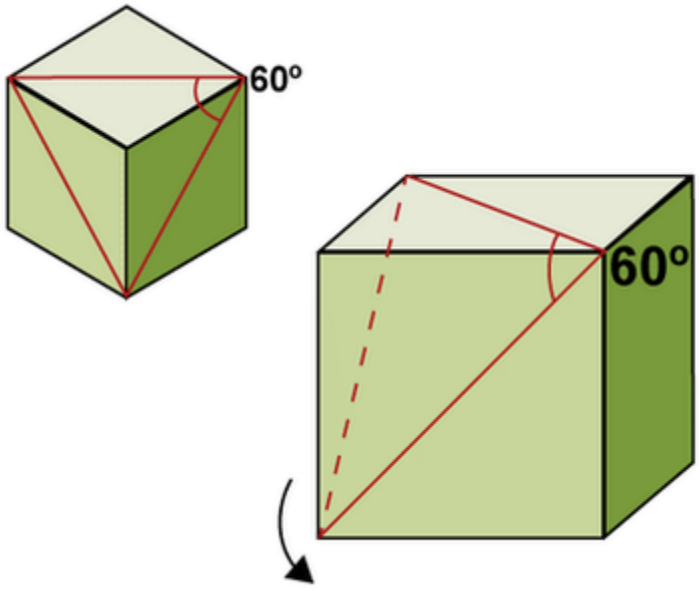
#2. What angle is made by the two red lines drawn on the two sides of the cube, as shown in the illustration?



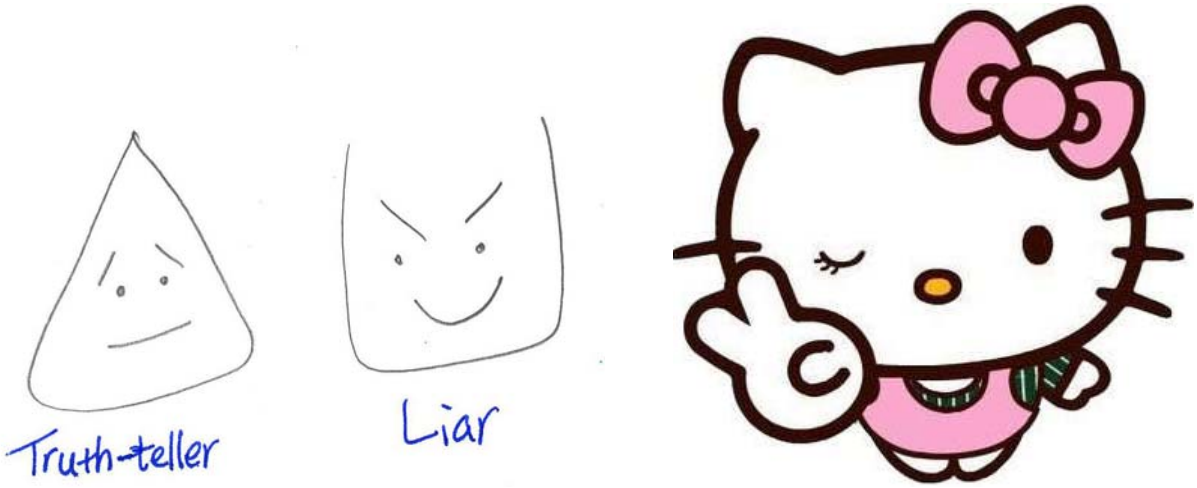
(a) 45° (b) 60° (c) 90° (d) ?

***source**: Martin Gardner (1914-2010)

Answer:



#3. Suppose you visit a distant planet inhabited by two groups of aliens, compulsive liars and faithful truth-tellers. You come to a fork in a road - one road goes to the left, the other to the right. You meet two aliens there, one a liar, the other a truth-teller... but you don't know which is which. You can ask just one YES/NO question to discover which road will take you to Hello Kitty. What question would you ask?

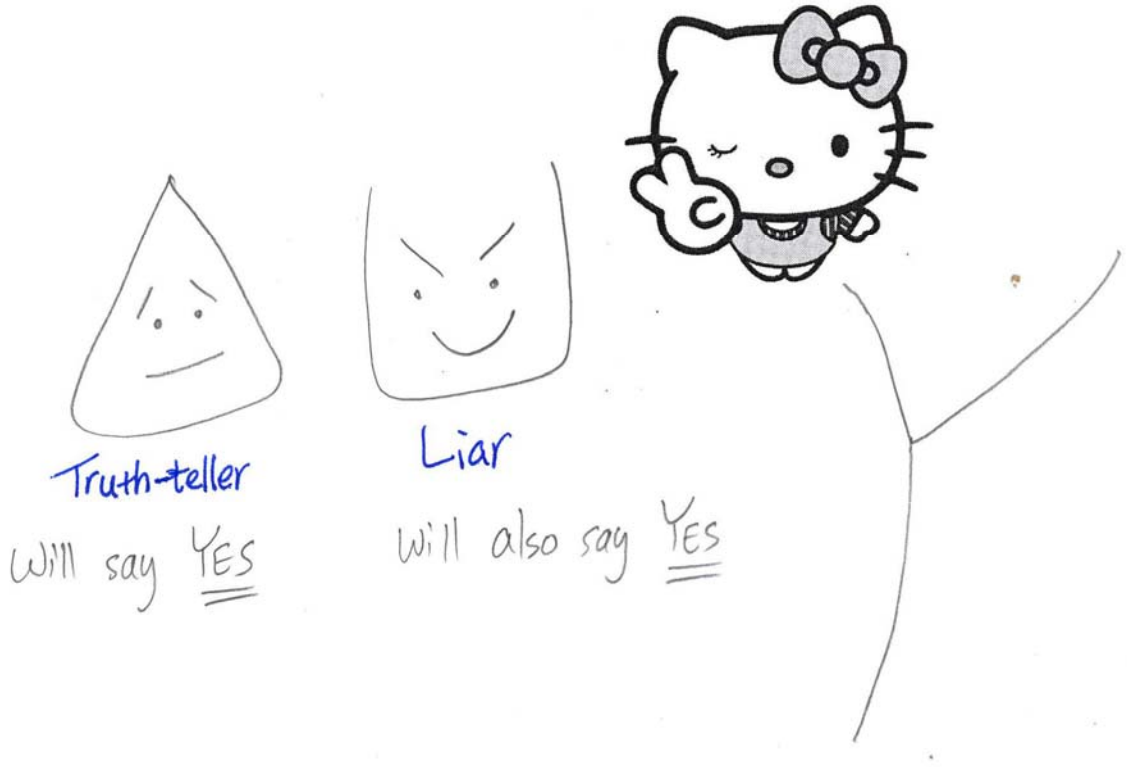


(a) Should I go to the right?
(b) Should I go to the left?
(c) Can you help me find Hello Kitty?
(d) ?

Answer: (Ask one while pointing at the other guy)
If I ask him if the road to the right goes to Hello Kitty, what would he say?

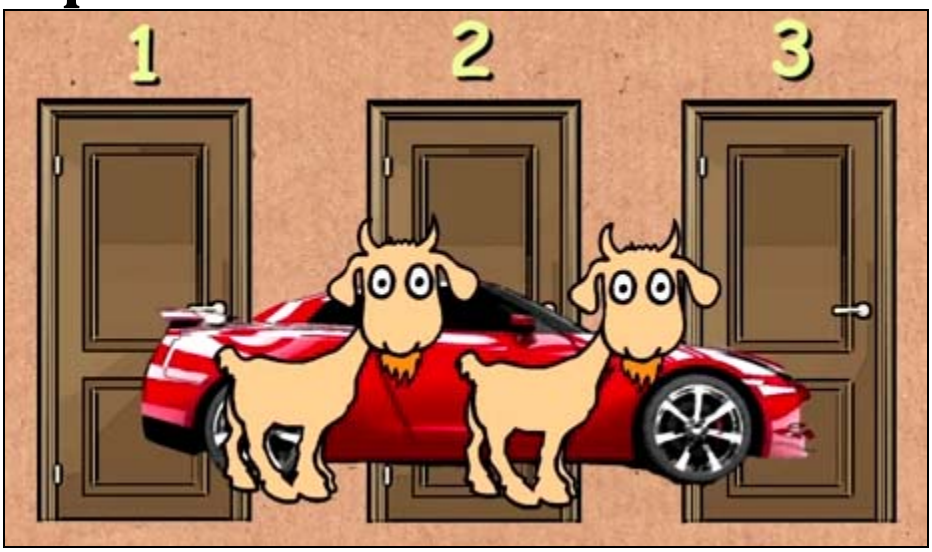
Case 1. Assume the road to the right does lead to Hello Kitty.

Case 2. Assume Hello Kitty is on the left.



#4. {The Monty Hall Problem}

There are 3 doors, behind which are two goats and an expensive car.



1. **Pick a door (Monty reveals one goat from the other two doors)**
2. **Stay or switch? Does it matter?**

Answer:

Surprisingly, the odds aren't 50-50. If you switch doors you'll win 2/3 of the time!

Fallen Ears from Introductory Statistics

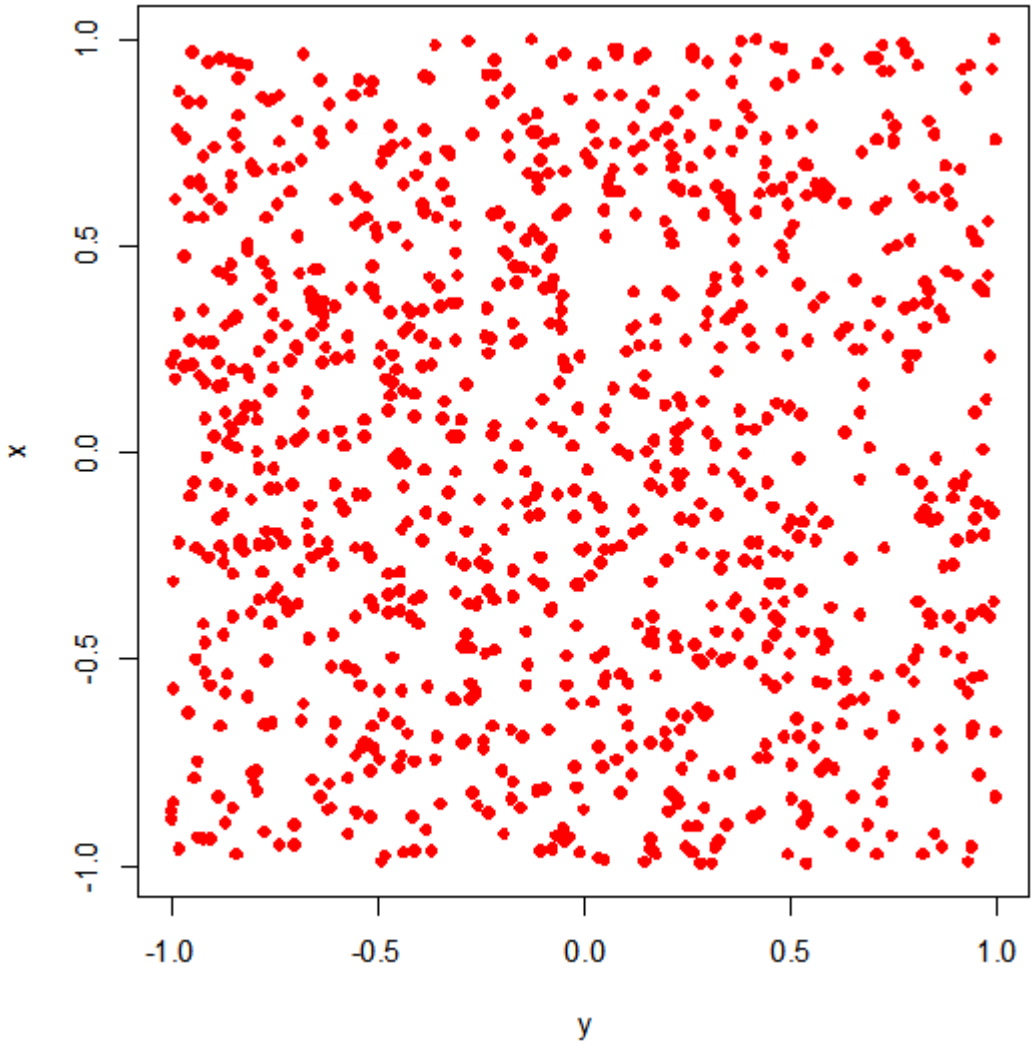
Feb. 12, 2015

Yoon G Kim

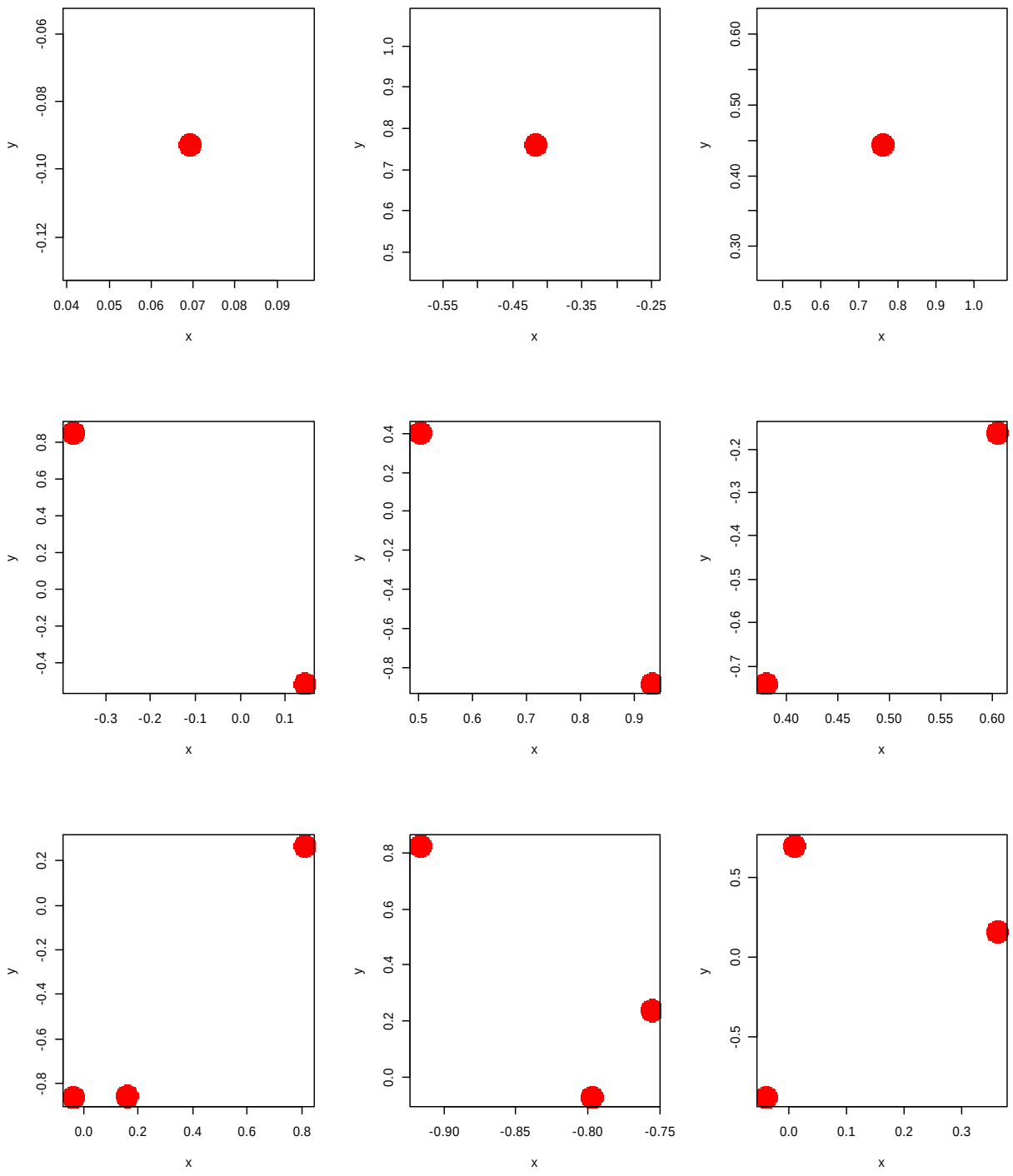
Dept. of Mathematics
Humboldt State Univ.
http://users.humboldt.edu/ygkim

[#1] Show me something "random."

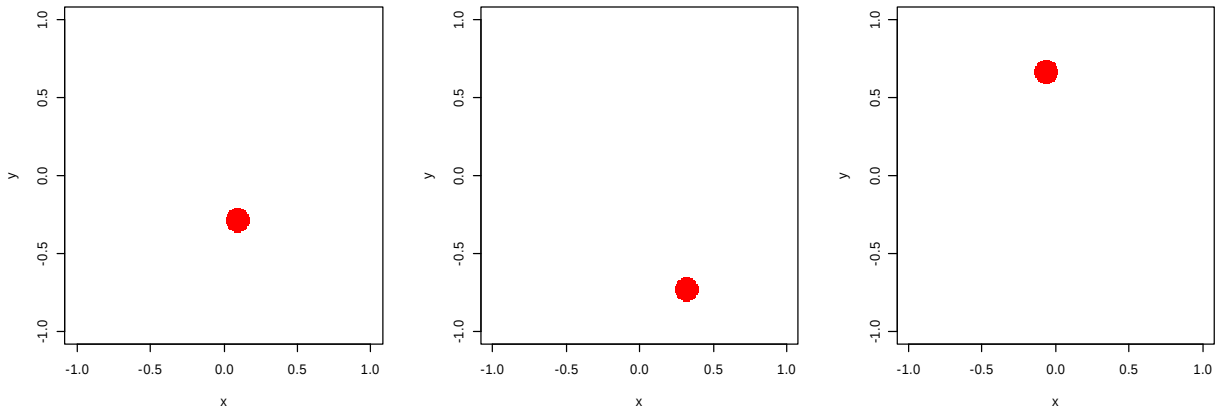
```
n = 1000
x <- runif(n, -1, 1)
y <- runif(n, -1, 1)
par(pin=c(5,5))
plot(y,x,pch=16,col="red")
```



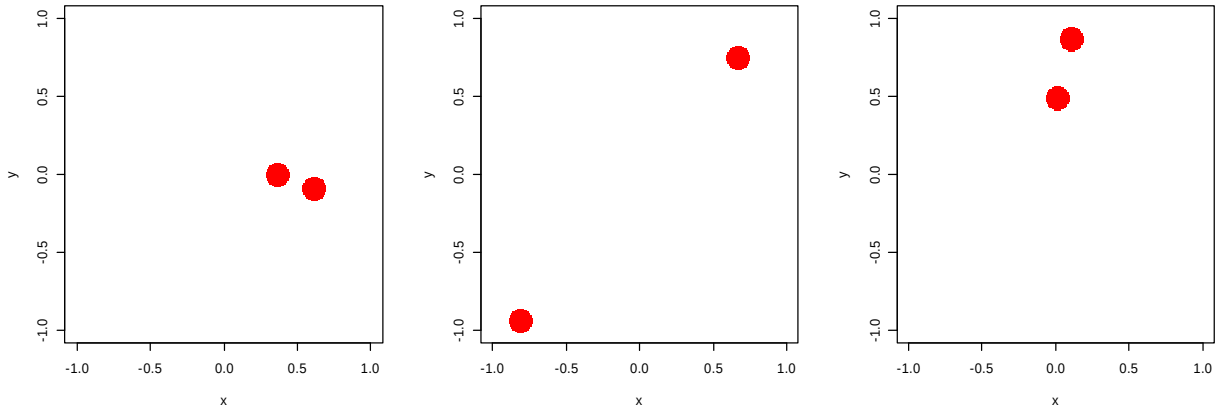
Oh, really? See this.

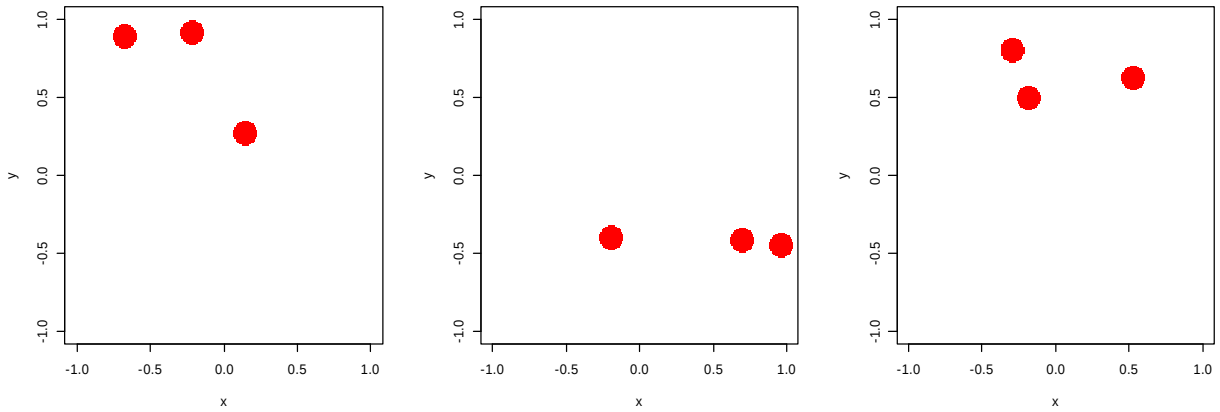


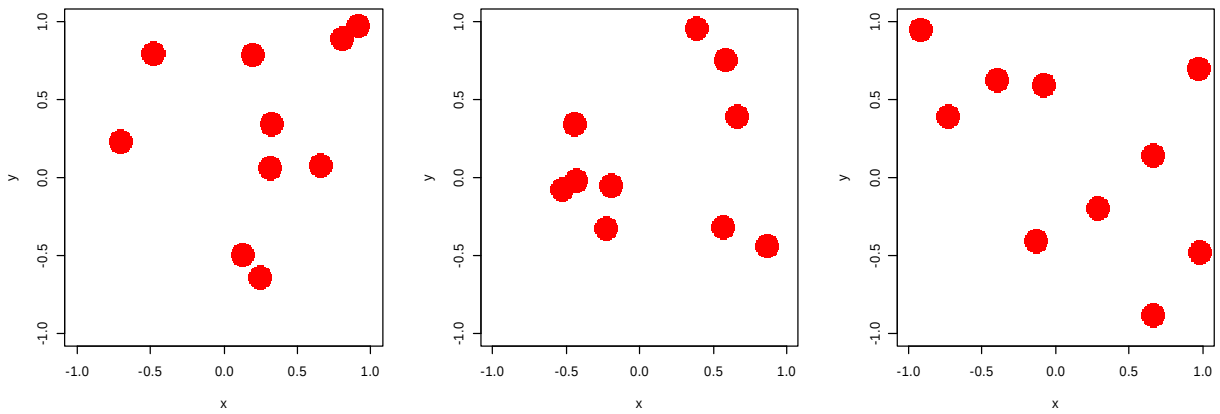
OMG, it always comes out like this! What's wrong?



Check the axes carefully.



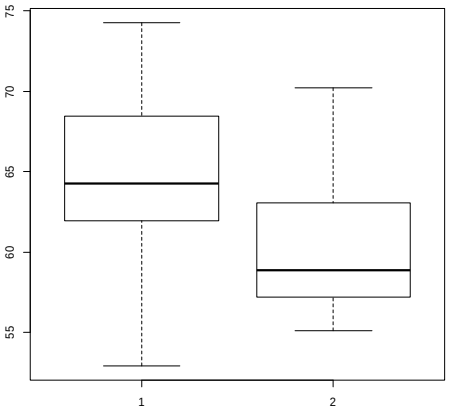




[#2] Averages 65 vs. 61. Are they "significantly" different?

<u>Short Answer</u>: It depends! Of course, sample size matters.

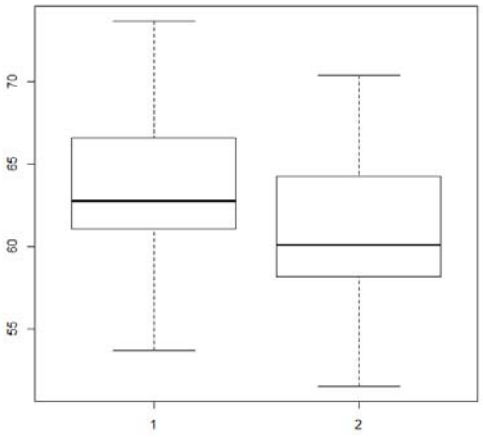
```
A10 <- rnorm(10,65,5)
B10 <- rnorm(10,61,5)
boxplot(A10,B10)
```




```
t.test(A10,B10,var.equal=T)
```

```
        Two Sample t-test

data:  A10 and B10
t = 1.5891, df = 18, p-value = 0.1294
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.258761  9.076062
sample estimates:
mean of x mean of y
 64.35120  60.44255
```

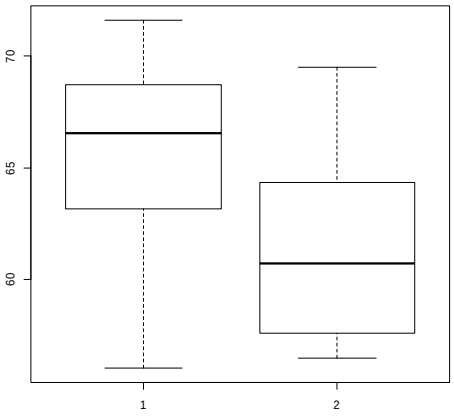
```
A15 <- rnorm(15,65,5)
B15 <- rnorm(15,61,5)
boxplot(A15,B15)
```




```
t.test(A15,B15,var.equal=T)
```

```
        Two Sample t-test

data:  A15 and B15
t = 1.4525, df = 28, p-value = 0.1575
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -1.153972  6.779573
sample estimates:
mean of x mean of y
 63.80399  60.99118
```

```
A20 <- rnorm(20,65,5)
B20 <- rnorm(20,61,5)
boxplot(A20,B20)
```




```
t.test(A20,B20,var.equal=T)
```

```
        Two Sample t-test

data:  A20 and B20
t = 3.1502, df = 38, p-value = 0.003175
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.509656 6.938820
sample estimates:
mean of x mean of y
 65.48318  61.25894
```

[#3] Percents 25% vs. 17% Are they "significantly" different?

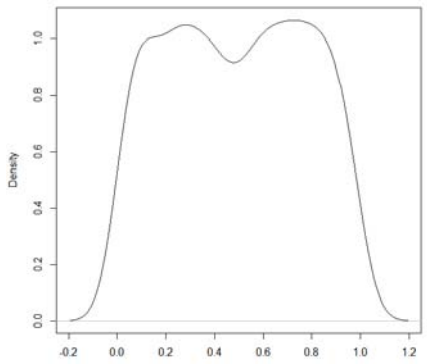
Short Answer: Same story. It depends! Of course, sample size matters.

```
sample1 <- matrix(c(25,17,75,83),nrow=2,ncol=2)
sample1
     [,1] [,2]
[1,]   25   75
[2,]   17   83
chisq.test(as.table(sample1))
        Pearson's Chi-squared test with Yates' continuity correction

data:  as.table(sample1)
X-squared = 1.4768, df = 1, p-value = 0.2243

sample2 <- matrix(c(125,85,375,415),nrow=2,ncol=2)
sample2
     [,1] [,2]
[1,]  125  375
[2,]   85  415
chisq.test(as.table(sample2))
        Pearson's Chi-squared test with Yates' continuity correction

data:  as.table(sample2)
X-squared = 9.1682, df = 1, p-value = 0.002463
```

[#4] What do you mean – conclusion at α=0.05?

Short Answer: Conclusion at 0.05 means that things do go the other way 5% of the times. Also, remember Murphy's Law.

```
myP <- numeric(1000)
for (i in 1:1000) {
   data1 <- rnorm(500)
   omg <- shapiro.test(data1)
   myP[i] <- omg$p }
summary(myP)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
0.0002126 0.2335000 0.4993000 0.4898000 0.7442000 0.9951000
plot(density(myP))
```

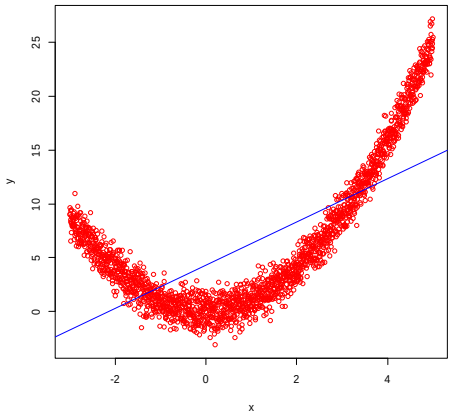



```
sum(myP < 0.05)
[1] 939
61/1000
[1] 0.061
```

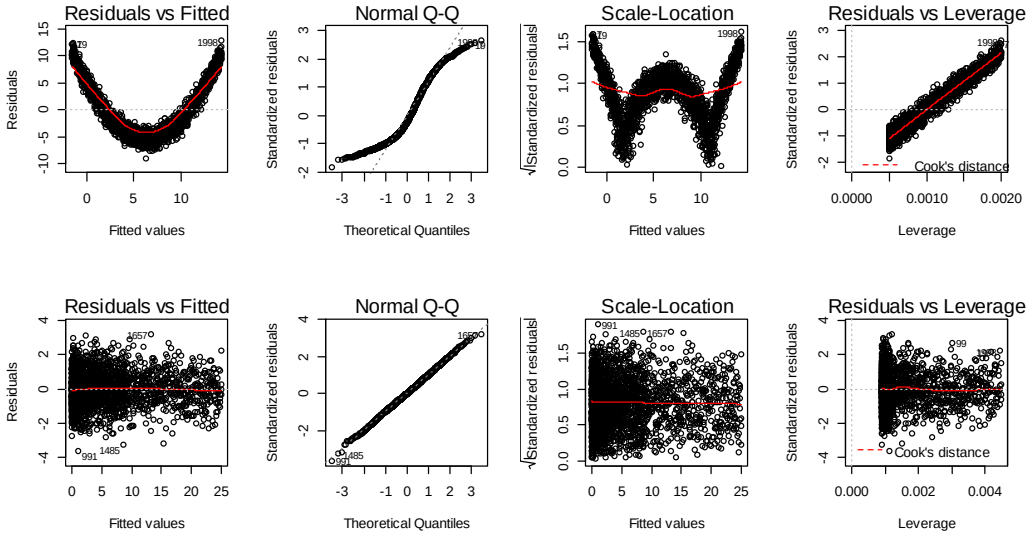
[#5] Why are the "residuals" so important in data analysis?

Short Answer: All the clues are in the "residuals."

```
x <- seq(-3,5,length=2000)
y <- x^2 + rnorm(2000)
plot(y~x)
model1 <- lm(y~x)
abline(model1)
```




```
library(faraway)
sumary(model1)
            Estimate Std. Error t value  Pr(>|t|)
(Intercept) 4.292743   0.119429  35.944 < 2.2e-16
x           2.008907   0.047436  42.349 < 2.2e-16
n = 2000, p = 2, Residual SE = 4.90166, R-Squared = 0.47

par(mfrow=c(2,4))
plot(model1)
model11 <- lm(y~x+I(x^2))
anova(model11)
Analysis of Variance Table

            Df Sum Sq Mean Sq F value    Pr(>F)
x            1  42296   42296   42146 < 2.2e-16 ***
I(x^2)       1  45672   45672   45511 < 2.2e-16 ***
Residuals 1997   2004       1
plot(model11)
```



[#6] When there is a significant "interaction," interpretation is NOT business as usual.

Demo:

```
data1 <- read.csv("U:\\STAT333\\TreeData.csv")
data1
```

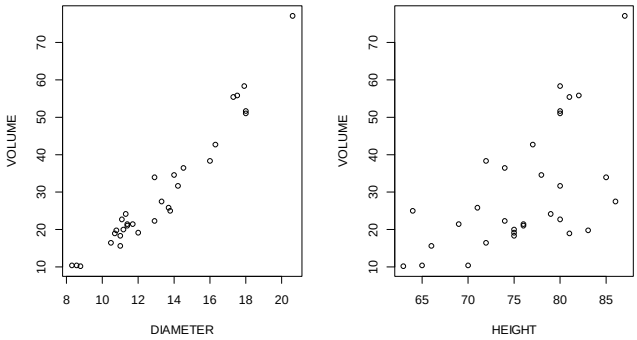
```
   DIAMETER HEIGHT VOLUME
1       8.3     70   10.3
2       8.6     65   10.3
3       8.8     63   10.2
4      10.5     72   16.4
5      10.7     81   18.8
6      10.8     83   19.7
7      11.0     66   15.6
8      11.0     75   18.2
9      11.1     80   22.6
10     11.2     75   19.9
11     11.3     79   24.2
12     11.4     76   21.0
13     11.4     76   21.4
14     11.7     69   21.3
15     12.0     75   19.1
16     12.9     74   22.2
17     12.9     85   33.8
18     13.3     86   27.4
19     13.7     71   25.7
20     13.8     64   24.9
21     14.0     78   34.5
22     14.2     80   31.7
23     14.5     74   36.3
24     16.0     72   38.3
25     16.3     77   42.6
26     17.3     81   55.4
27     17.5     82   55.7
28     17.9     80   58.3
29     18.0     80   51.5
30     18.0     80   51.0
31     20.6     87   77.0
```

```
attach(data1)
model1 <- lm(VOLUME~DIAMETER*HEIGHT)
sumary(model1)
```

```
                Estimate Std. Error t value  Pr(>|t|)
(Intercept)     69.396316  23.835755  2.9114 0.0071307
DIAMETER        -5.855848   1.921336 -3.0478 0.0051087
HEIGHT          -1.297083   0.309843 -4.1863 0.0002699
DIAMETER:HEIGHT  0.134654   0.024377  5.5238 7.484e-06

n = 31, p = 4, Residual SE = 2.70855, R-Squared = 0.98
```

Q: OMG, why are the coefficients minus??? Thicker trees, taller trees have smaller Volume???

```
par(mfrow=c(1,2))
plot(VOLUME~DIAMETER)
plot(VOLUME~HEIGHT)
```



Let's check the estimated V for D=10 & H=75

```
69.39632+(-5.85585*10)+(-1.29708*75)+(0.13465*10*75)
[1] 14.54432
```

Estimated V for D=15 & H=85

```
69.39632+(-5.85585*15)+(-1.29708*85)+(0.13465*15*85)
[1] 42.98552
```

[#7] Is anything going to change if "standardized" data are used?

Short Answer: YEP! Night and Day!

```
data2 <- data.frame(scale(data1))
model2 <- lm(VOLUME~DIAMETER*HEIGHT,data2)
sumary(model2)
                Estimate Std. Error t value  Pr(>|t|)
(Intercept)     -0.082314   0.033135 -2.4842   0.01948
DIAMETER         0.835779   0.037006 22.5847 < 2.2e-16
HEIGHT           0.188726   0.036695  5.1432 2.073e-05
DIAMETER:HEIGHT  0.163799   0.029654  5.5238 7.484e-06

n = 31, p = 4, Residual SE = 0.16478, R-Squared = 0.98

summary(model1)
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     69.39632   23.83575   2.911  0.00713 **
DIAMETER        -5.85585    1.92134  -3.048  0.00511 **
HEIGHT          -1.29708    0.30984  -4.186  0.00027 ***
DIAMETER:HEIGHT  0.13465    0.02438   5.524 7.48e-06 ***
---
Residual standard error: 2.709 on 27 degrees of freedom
Multiple R-squared:  0.9756,    Adjusted R-squared:  0.9728
F-statistic: 359.3 on 3 and 27 DF,  p-value: < 2.2e-16

summary(model2)
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.08231    0.03313  -2.484   0.0195 *
DIAMETER         0.83578    0.03701  22.585  < 2e-16 ***
HEIGHT           0.18873    0.03669   5.143 2.07e-05 ***
DIAMETER:HEIGHT  0.16380    0.02965   5.524 7.48e-06 ***
---
Residual standard error: 0.1648 on 27 degrees of freedom
Multiple R-squared:  0.9756,    Adjusted R-squared:  0.9728
F-statistic: 359.3 on 3 and 27 DF,  p-value: < 2.2e-16
```

[#8] See the problems with AIC, BIC, etc.

```
AIC(model1,model2)
```

```
       df       AIC
model1  5 155.46916
model2  5 -18.10519

BIC(model1,model2)
       df       BIC
model1  5 162.63910
model2  5 -10.93526
```

[#9] *lm* vs. *glm*, Are the two results very different?

Short Answer: Usually not much. But it can be quite different in "extreme" cases.
Advice: Try both to gain more insight.

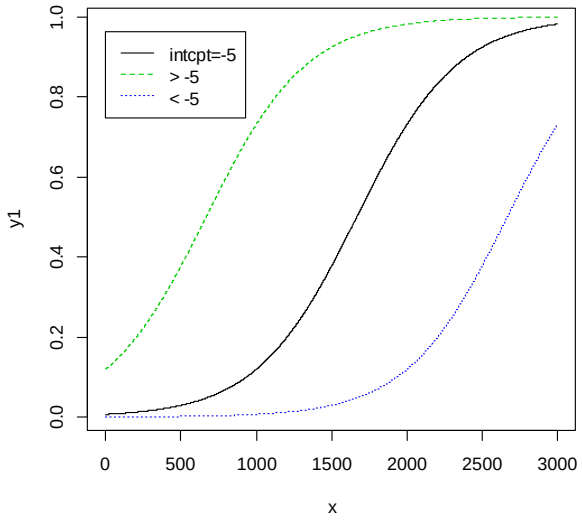
```
head(pima)
  pregnant glucose diastolic triceps insulin  bmi diabetes age test
1        6     148        72      35       0 33.6    0.627  50    1
2        1      85        66      29       0 26.6    0.351  31    0
3        8     183        64       0       0 23.3    0.672  32    1
4        1      89        66      23      94 28.1    0.167  21    0
5        0     137        40      35     168 43.1    2.288  33    1
6        5     116        74       0       0 25.6    0.201  30    0
table(pima$test)

  0   1
500 268

model1 <- lm(test~pregnant+glucose+diastolic+triceps+insulin+bmi+diabetes+age, data=pima)

model2 <- glm(test~pregnant+glucose+diastolic+triceps+insulin+bmi+diabetes+age,
family=binomial,data=pima)

summary(model1)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.8538943  0.0854850  -9.989  < 2e-16 ***
pregnant     0.0205919  0.0051300   4.014 6.56e-05 ***
glucose      0.0059203  0.0005151  11.493  < 2e-16 ***
diastolic   -0.0023319  0.0008116  -2.873  0.00418 **
triceps      0.0001545  0.0011122   0.139  0.88954
insulin     -0.0001805  0.0001498  -1.205  0.22857
bmi          0.0132440  0.0020878   6.344 3.85e-10 ***
diabetes     0.1472374  0.0450539   3.268  0.00113 **
age          0.0026214  0.0015486   1.693  0.09092 .
---
Residual standard error: 0.4002 on 759 degrees of freedom
Multiple R-squared:  0.3033,    Adjusted R-squared:  0.2959
F-statistic: 41.29 on 8 and 759 DF,  p-value: < 2.2e-16

summary(model2)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.4046964  0.7166359 -11.728  < 2e-16 ***
pregnant     0.1231823  0.0320776   3.840 0.000123 ***
glucose      0.0351637  0.0037087   9.481  < 2e-16 ***
diastolic   -0.0132955  0.0052336  -2.540 0.011072 *
triceps      0.0006190  0.0068994   0.090 0.928515
```

```
insulin     -0.0011917  0.0009012  -1.322 0.186065
bmi          0.0897010  0.0150876   5.945 2.76e-09 ***
diabetes     0.9451797  0.2991475   3.160 0.001580 **
age          0.0148690  0.0093348   1.593 0.111192
---
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 993.48  on 767  degrees of freedom
Residual deviance: 723.45  on 759  degrees of freedom
AIC: 741.45

Number of Fisher Scoring iterations: 5

AIC(model1,model2)
       df      AIC
model1 10 783.8218
model2  9 741.4454

BIC(model1,model2)
       df      BIC
model1 10 830.2597
model2  9 783.2395
```

[#10] What roles do α and β play in the logistic model $y=1/\{1+e^{-(\alpha+\beta x)}\}$?

Intercept controls the value of *x* (i.e., make "events" earlier or later) for a given probability.

Exhibit 1:
Three different logistic plots with the same slopes, but with different intercepts.



R codes to produce this:

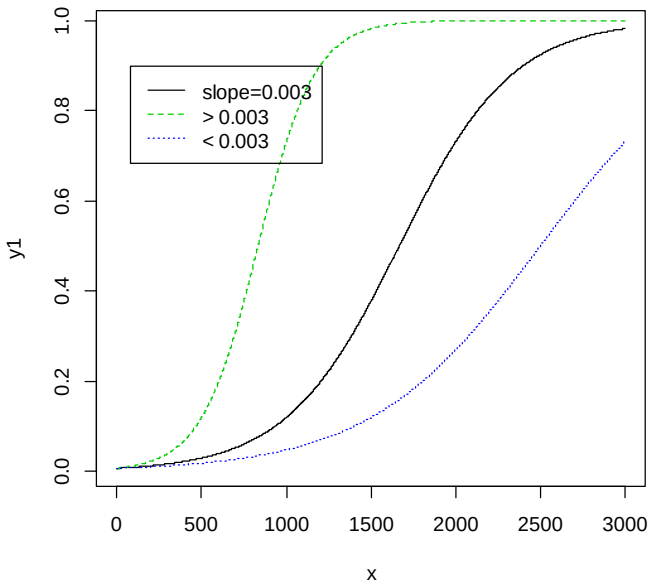
```
x <- seq(0,3000,0.2)
y1 <- 1/(1+exp(-(-5+(0.003*x))))
plot(x,y1,type="l")
y2 <- 1/(1+exp(-(-2+(0.003*x))))
lines(x,y2,col=3, lty=2)
y3 <- 1/(1+exp(-(-8+(0.003*x))))
lines(x,y3,col=4, lty=3)
legend(locator(1),c("intcpt=-5","> -
5","< -5"),lty=1:3,col=c(1,3,4))
```

Slope controls the steepness of the curve (i.e., make probability of events) steeper or less for a given *x* (i.e., date).

Exhibit 2:
Three different logistc plots with the same intercepts, but with different slopes.

R codes to produce this:

```
x <- seq(0,3000,0.2)
y1 <- 1/(1+exp(-(-5+(0.003*x))))
plot(x,y1,type="l")
y2 <- 1/(1+exp(-(-5+(0.006*x))))
lines(x,y2,col=3, lty=2)
y3 <- 1/(1+exp(-(-5+(0.002*x))))
lines(x,y3,col=4, lty=3)
legend(locator(1),c("slope=0.003","
0.003","< 0.003"),lty=1:3,col=c(1,3,4))
```


slope=0.003
> 0.003
< 0.003
y1
x
0.0
0.2
0.4
0.6
0.8
1.0
0
500
1000
1500
2000
2500
3000