StatR 101: Fall 2012

Homework 5

Rod Doe

Saturday, October 27, 2012

# Summarizing Summary Statistics

## Function SummaryStatsA

SummaryStatsA <- function(x, y) {

summaryNames <- c("x.bar", "y.bar", "x.sd", "y.sd", "r", "a", "b", "ss.total", "ss.model", "ss.residual", "r2")

n = length(x)

x.bar = sum(x) / n

y.bar = sum(y) / n

x.var = sum((x - x.bar)^2) / (n - 1)

y.var = sum((y - y.bar)^2) / (n - 1)

x.sd = sqrt(x.var)

y.sd = sqrt(y.var)

ss.x = sum(x^2) - ((sum(x)^2) / n)

ss.y = sum(y^2) - ((sum(y)^2) / n)

ss.xy = sum(x \* y) - ((sum(x) \* sum(y)) / n)

b = ss.xy / ss.x

a = (sum(y) / n) - (b \* sum(x) / n)

y.hat = a + x \* b

r = ss.xy / sqrt(ss.x \* ss.y)

# SStotal = SSmodel + SSerror

ss.total = sum((y - y.bar)^2)

# SSmodel = sum((y.hat - y.bar)^2)

ss.model = sum((y.hat - y.bar)^2)

# SSresidual = sum((y - y.hat)^2)

ss.residual = sum((y-y.hat)^2)

r2 = r \* r

result = c(x.bar, y.bar, x.sd, y.sd, r, a, b, ss.total, ss.model, ss.residual, r2)

names(result) = summaryNames

return(result)

}

## Graphics from SummaryStatsA

Data source: ‘Statistics – An Introduction Using R’, Michael J. Crawley, ISBN 0-470-02298-1

I plagiarized this simple data set from the Crawley text so that it would be easy to verify the results.

tannin = c(0:8)

growth = c(12,10,8,11,6,7,2,3,3)

par(mfrow=c(2,2))

plot(tannin, growth, pch=16, main="Growth = f(tannin)")

abline(lm(growth~tannin))

plot(tannin, growth, pch=16, main="SStotal")

abline(lm(growth~tannin))

abline(h=mean(growth), col=rgb(1,0,0))

fitted=predict(lm(growth~tannin))

for (i in 1:length(tannin)) lines( c(tannin[i],tannin[i]), c(growth[i],mean(growth)), col=rgb(1,0,0) )

plot(tannin, growth, pch=16, main="SSmodel")

abline(lm(growth~tannin))

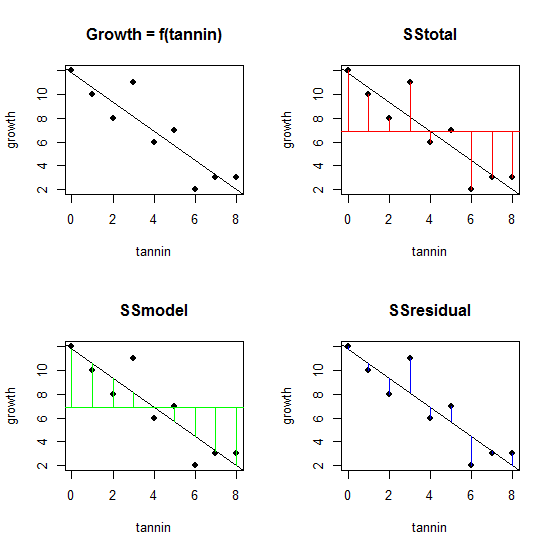
abline(h=mean(growth), col=rgb(0,1,0))

for (i in 1:length(tannin)) lines( c(tannin[i],tannin[i]), c(fitted[i],mean(growth)), col=rgb(0,1,0) )

plot(tannin, growth, pch=16, main="SSresidual")

abline(lm(growth~tannin))

for (i in 1:length(tannin)) lines( c(tannin[i],tannin[i]), c(fitted[i],growth[i]), col=rgb(0,0,1) )



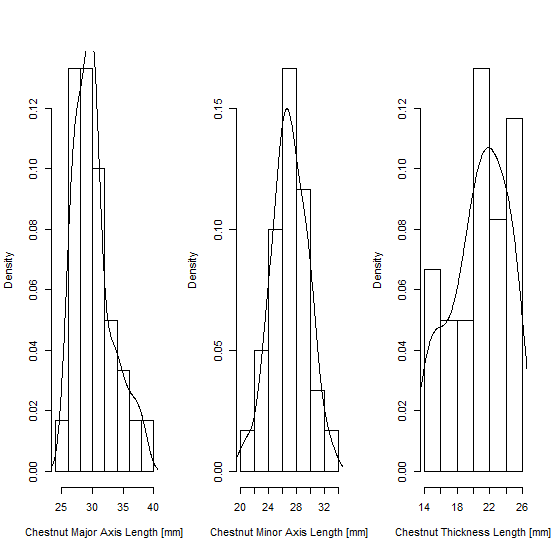
These data were too easy. I wanted to go out into the wild and capture some data of my own, so I trekked through the neighborhood in search of data. I found a chestnut tree that was surrounded by chestnuts on the ground. I noticed that each chestnut had these easily measurable attributes:

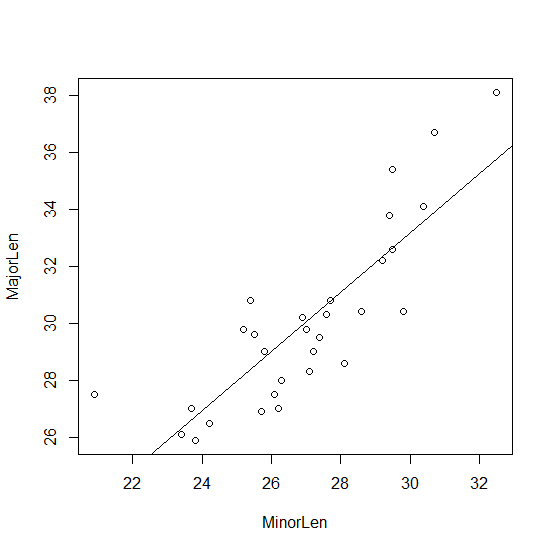
* Major axis length
* Minor axis length
* Thickness

So, I grabbed a random sample of 30 chestnuts, took out my trusty micrometer, and made some data.

NOTE: Upon conclusion of data acquisition, the chestnuts used in this study were released back into the wild.







The data indicated that the major length could be modeled as a linear function of the minor length with r = 0.84, a = 1.986, b = 1.039

cor(MajorLen,MinorLen)

[1] 0.8411849

lm(MajorLen~MinorLen)

Call:

lm(formula = MajorLen ~ MinorLen)

Coefficients:

(Intercept) MinorLen

1.986 1.039

## Function SummaryStatsB

SummaryStatsB <- function(x, y) {

summaryNames <- c("x.bar", "y.bar", "x.sd", "y.sd", "r", "a", "b", "ss.total", "ss.model", "ss.residual", "r2")

n = length(x)

x.bar = mean(x)

y.bar = mean(y)

x.sd = sd(x)

y.sd = sd(y)

m = lm(y~x)

a = m$coefficients[1]

b = m$coefficients[2]

y.hat = m$fitted.values

r = cor(x,y)

# SStotal = SSmodel + SSerror

ss.total = var(y) \* (n - 1)

# SSmodel = sum((y.hat - y.bar)^2)

ss.model = var(y.hat) \* (n - 1)

# SSresidual = sum((y - y.hat)^2)

ss.residual = ss.total - ss.model

r2 = r \* r

result = c(x.bar, y.bar, x.sd, y.sd, r, a, b, ss.total, ss.model, ss.residual, r2)

names(result) = summaryNames

return(result)

}

## Results Comparision

SummaryStatsB yields identical results to SummaryStatsA:

SummaryStatsB(x,y)

x.bar y.bar x.sd y.sd r a

4.0000000 6.8888889 2.7386128 3.6893239 -0.9031408 11.7555556

b ss.total ss.model ss.residual r2

-1.2166667 108.8888889 88.8166667 20.0722222 0.8156633

SummaryStatsA(x,y)

x.bar y.bar x.sd y.sd r a

4.0000000 6.8888889 2.7386128 3.6893239 -0.9031408 11.7555556

b ss.total ss.model ss.residual r2

-1.2166667 108.8888889 88.8166667 20.0722222 0.8156633

# Anscombe Data Analysis

I loaded the anscombe data set and ran them through function SummaryStatsB. The data appear to be nearly identical, with nearly identical values of x.bar, y.bar, x.sd, y.sd, a, b, ss.total, ss.model, and ss.residual.

> SummaryStatsB(anscombe$x1, anscombe$y1)

x.bar y.bar x.sd y.sd r a

9.0000000 7.5009091 3.3166248 2.0315681 0.8164205 3.0000909

b ss.total ss.model ss.residual r2

0.5000909 41.2726909 27.5100009 13.7626900 0.6665425

> SummaryStatsB(anscombe$x2, anscombe$y2)

x.bar y.bar x.sd y.sd r a

9.0000000 7.5009091 3.3166248 2.0316567 0.8162365 3.0009091

b ss.total ss.model ss.residual r2

0.5000000 41.2762909 27.5000000 13.7762909 0.6662420

> SummaryStatsB(anscombe$x3, anscombe$y3)

x.bar y.bar x.sd y.sd r a

9.0000000 7.5000000 3.3166248 2.0304236 0.8162867 3.0024545

b ss.total ss.model ss.residual r2

0.4997273 41.2262000 27.4700082 13.7561918 0.6663240

> SummaryStatsB(anscombe$x4, anscombe$y4)

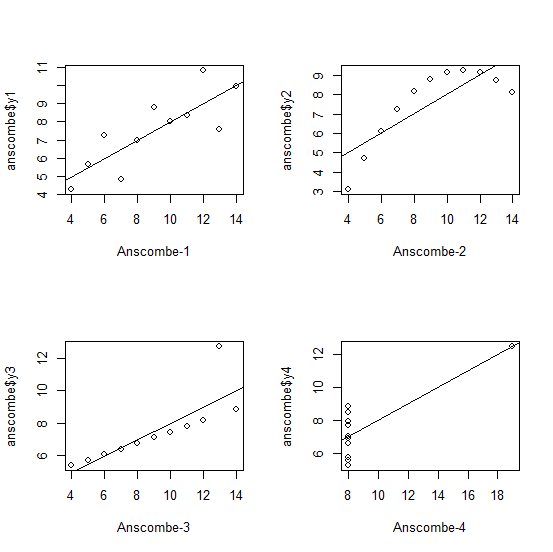
x.bar y.bar x.sd y.sd r a

9.0000000 7.5009091 3.3166248 2.0305785 0.8165214 3.0017273

b ss.total ss.model ss.residual r2

0.4999091 41.2324909 27.4900009 13.7424900 0.6667073

However, the plots of the four data pairs reveal some trickery:



The linear model is applicable for the first data set, but not for the other data sets.

When I consider the summary statistics reported, which indicated that these data should be quite similar, and then consider the plots of the actual data, which reveal that the data were contrived to yield identical summary statistics, I am reminded of Mariner coach Lou Piniella’s quote about statistics:

"Statistics are like bikinis -- they show a lot but not everything."

The take-home message here is that statistics will not tell the entire story. They measure, they summarize, but a plot is maybe worth a thousand statistics. Dr. Anscombe maybe had too much free time…

While researching Lou Piniella quotes, I found another that was quite amusing:

"We're fine. The only time we lose our concentration is when the umpire says, 'Play Ball.'"

# Bonus – Model the Anscombe data with a quadratic function

This did not work. Attempts to model as a quadratic function:

lm(y~x + I(x^2))

produced warning messages.

Warning message:

In abline(lm(anscombe$y1 ~ anscombe$x1 + I(anscombe$x1^2))) :

only using the first two of 3 regression coefficients

> par(mfrow=c(2,2))

> plot(anscombe$x1, anscombe$y1,xlab="anscombe-1")

> abline(lm(anscombe$y1 ~ anscombe$x1 + I(anscombe$x1^2)))

Warning message:

In abline(lm(anscombe$y1 ~ anscombe$x1 + I(anscombe$x1^2))) :

only using the first two of 3 regression coefficients

> plot(anscombe$x2, anscombe$y2,xlab="anscombe-2")

> abline(lm(anscombe$y2 ~ anscombe$x2 + I(anscombe$x2^2)))

Warning message:

In abline(lm(anscombe$y2 ~ anscombe$x2 + I(anscombe$x2^2))) :

only using the first two of 3 regression coefficients

> plot(anscombe$x3, anscombe$y3,xlab="anscombe-3")

> abline(lm(anscombe$y3 ~ anscombe$x3 + I(anscombe$x3^2)))

Warning message:

In abline(lm(anscombe$y3 ~ anscombe$x3 + I(anscombe$x3^2))) :

only using the first two of 3 regression coefficients

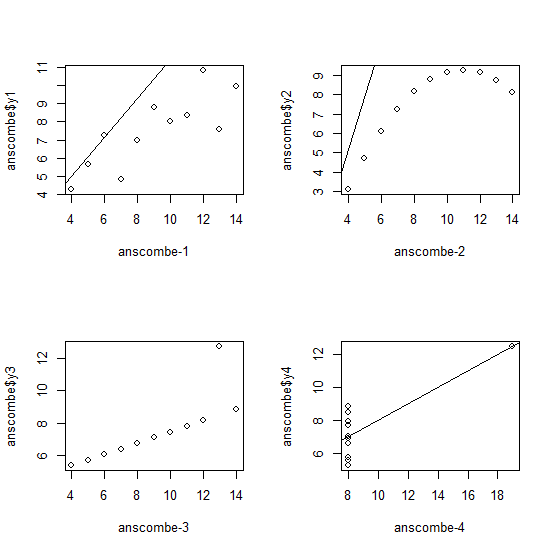
> plot(anscombe$x4, anscombe$y4,xlab="anscombe-4")

> abline(lm(anscombe$y4 ~ anscombe$x4 + I(anscombe$x4^2)))

Warning message:

In abline(lm(anscombe$y4 ~ anscombe$x4 + I(anscombe$x4^2))) :

only using the first two of 3 regression coefficients



This seemed like Anscombe’s data was confusing the lm function. To better understand this, I created some simple quadratic data and then plotted it:

> x = c(0:10)

> y = 2 \* x^2 + 3 \* x + 4

> x

[1] 0 1 2 3 4 5 6 7 8 9 10

> y

[1] 4 9 18 31 48 69 94 123 156 193 234

> plot(x, y)

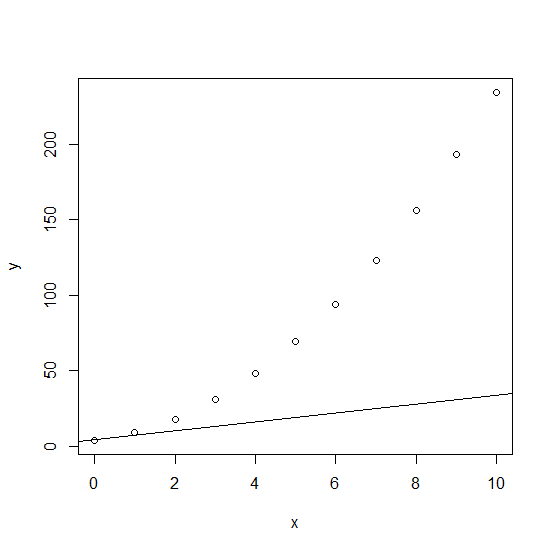
> abline(lm(y ~ x + I(x^2)))

Warning message:

In abline(lm(y ~ x + I(x^2))) :

only using the first two of 3 regression coefficients

The simple data produced the same warnings, and didn’t do what I had hoped it would.



Sigh - I will be interested to see the solution to this.