StatR 301: Spring 2013

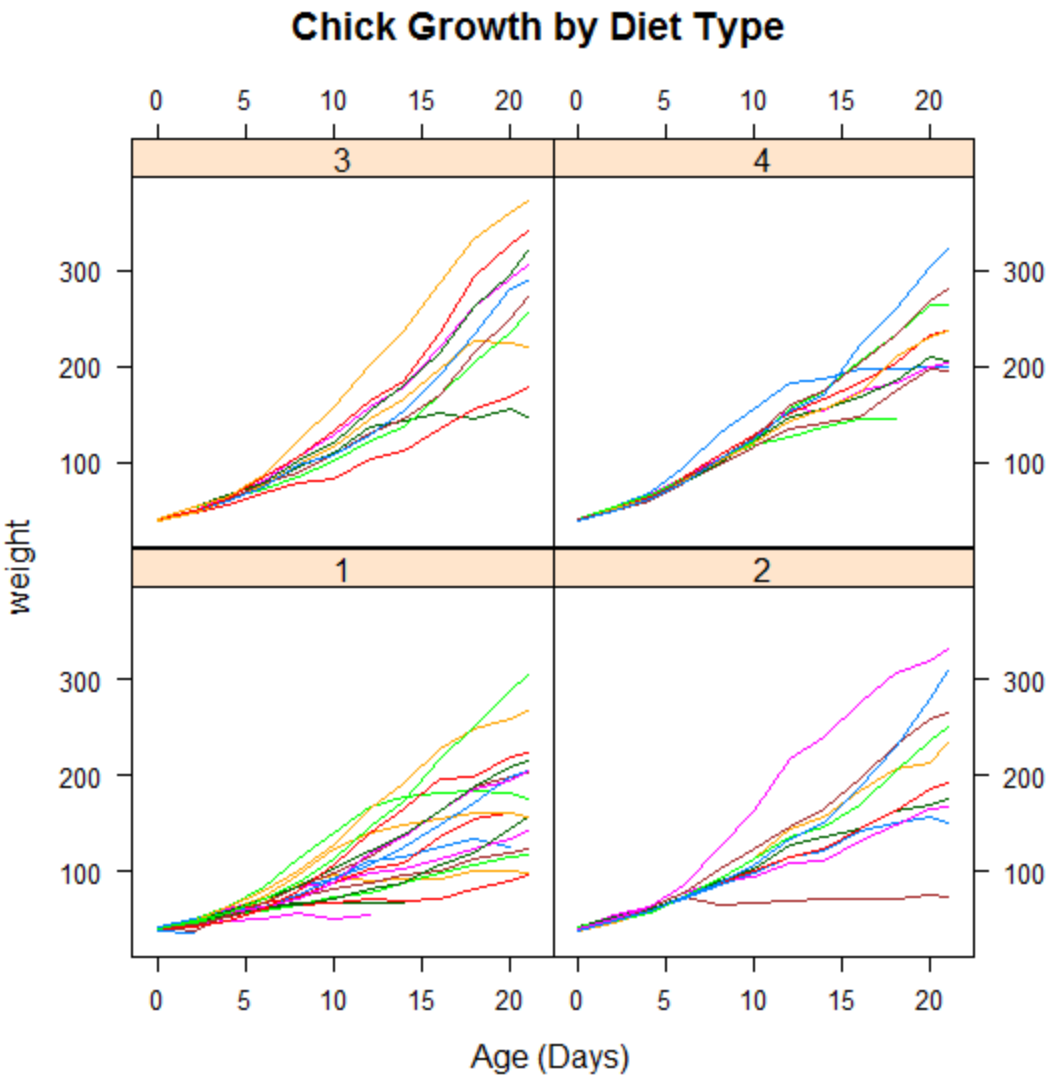
Homework 03

Rod Doe

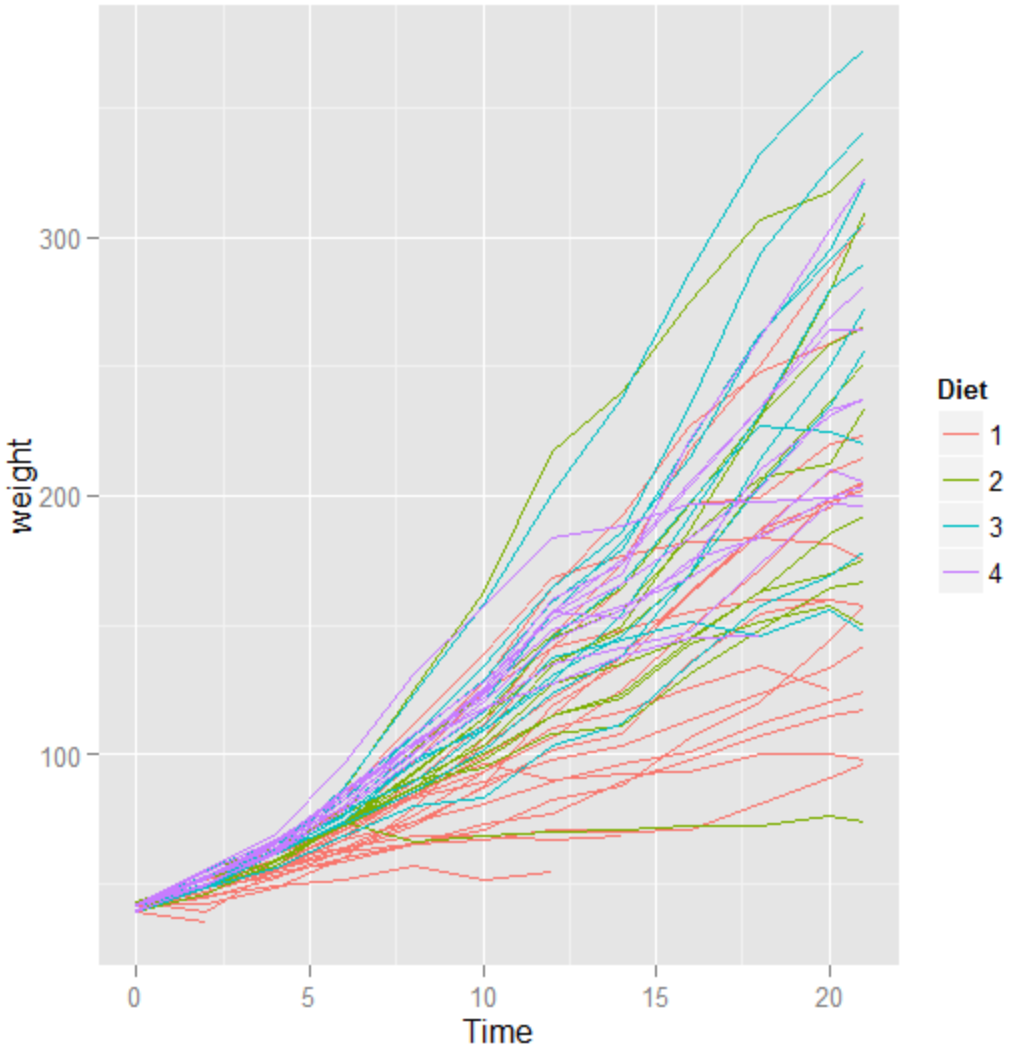
Thursday, May 9, 2013

My First ggplot

```
head(ChickWeight)
  # weight Time Chick Diet
# 1     42    0     1    1
# 2     51    2     1    1
# 3     59    4     1    1
# 4     64    6     1    1
# 5     76    8     1    1
# 6     93   10     1    1

# Assaf's example spaghetti plot.
library(lattice)
xyplot(weight~Time | Diet,data=ChickWeight,group=Chick,type='l',
      scales=list(alternating=3),main="Chick Growth by Diet Type",xlab="Age (Days)")
```



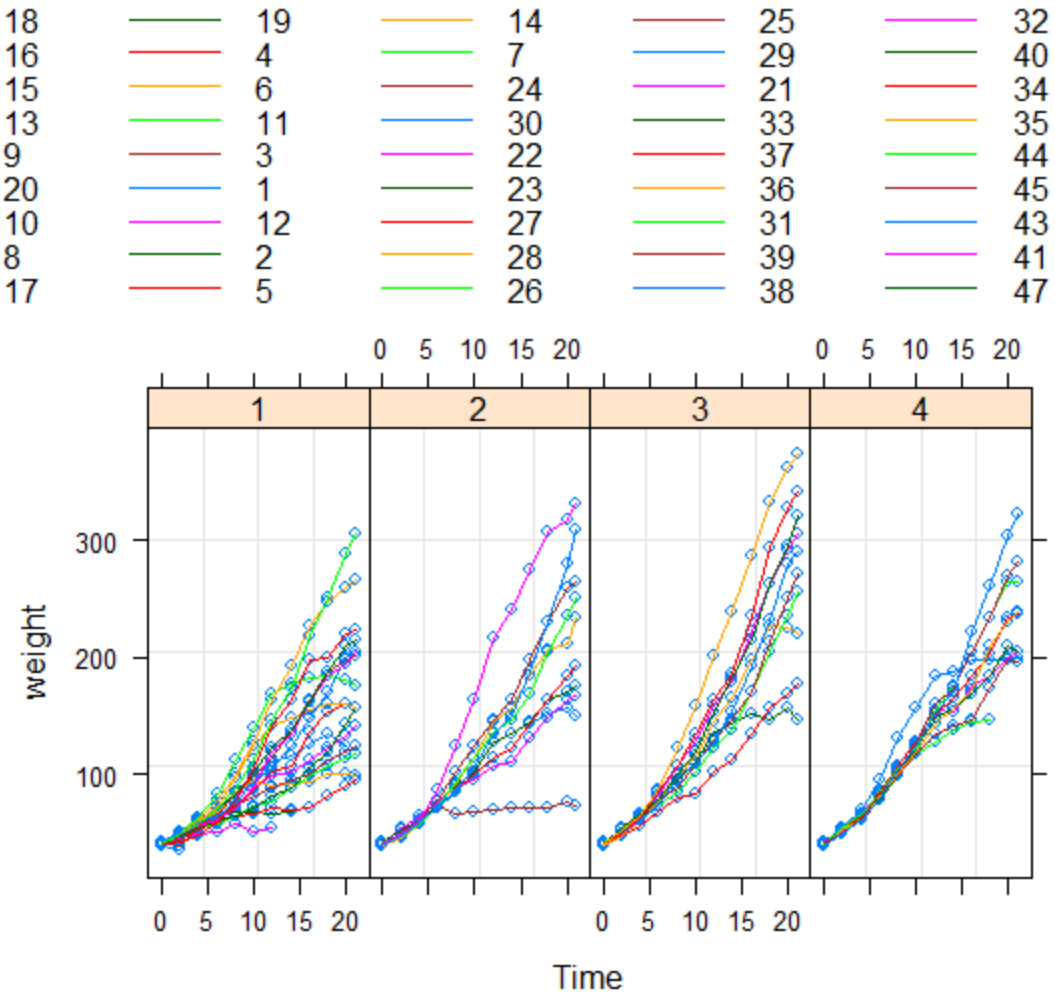

```
# Try to do the same thing with ggplot2
library(ggplot2)
qplot(x=Time, y=weight, data=ChickWeight, group=Chick, color=Diet, geom="line")
```



I could not find a way to split this into panes. The documentation of this function appeared to be written by somebody in a big hurry.

I started to play with the groupedData function, and found another way to render this with lattice.

```
library(lattice)

results <- groupedData(weight ~ Time |Chick, outer= ~ Diet,
order.groups=TRUE, data=ChickWeight)
plot(results, outer=TRUE)  # Useful and interesting.  Shows that 3 has
the most effect.
```



From a documentation standpoint, core R graphics beat lattice (Crawley's 'The R Book' has decent coverage of lattice), and lattice beats ggplot, whose documentation appears to be its Achilles Heel. Oh, but there are books available to explain these is more detail. The powerful capabilities of ggplot are only as good as their documentation.

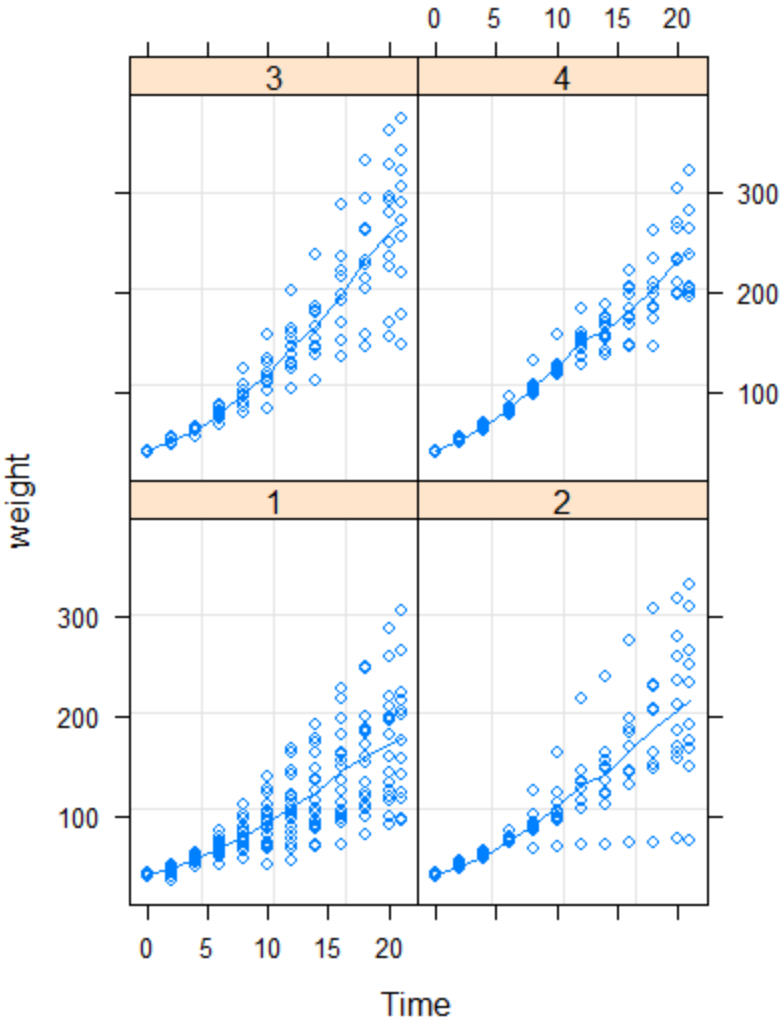
Mixed Effects

Mixed effects include fixed effects and random effects. Fixed effects influence the mean of y, whereas random effects influence the variation of y.

```
# Mixed effect model for ChickWeight
library(nlme)
```

It's probably not necessary/helpful to see information about each specific chick. Perhaps the data can be grouped differently to make this possible. I tried different things with this, and got R's version of the PC LOAD LETTER, which I buried with a suppressWarnings call.

```
suppressWarnings(     results <- groupedData(weight ~ Time | Diet,
outer= ~ Chick, order.groups=TRUE, data=ChickWeight)  )

suppressWarnings( plot(results) )  # Useful and interesting, shows
that 3 has the most effect (steepest slope for group).
```



This is better. It is possible to see that diet 3 has the largest slope, and hence the most effect. I wish I new about this plot when I was using R in the semiconductor industry. To see the signal in the noise is nice.

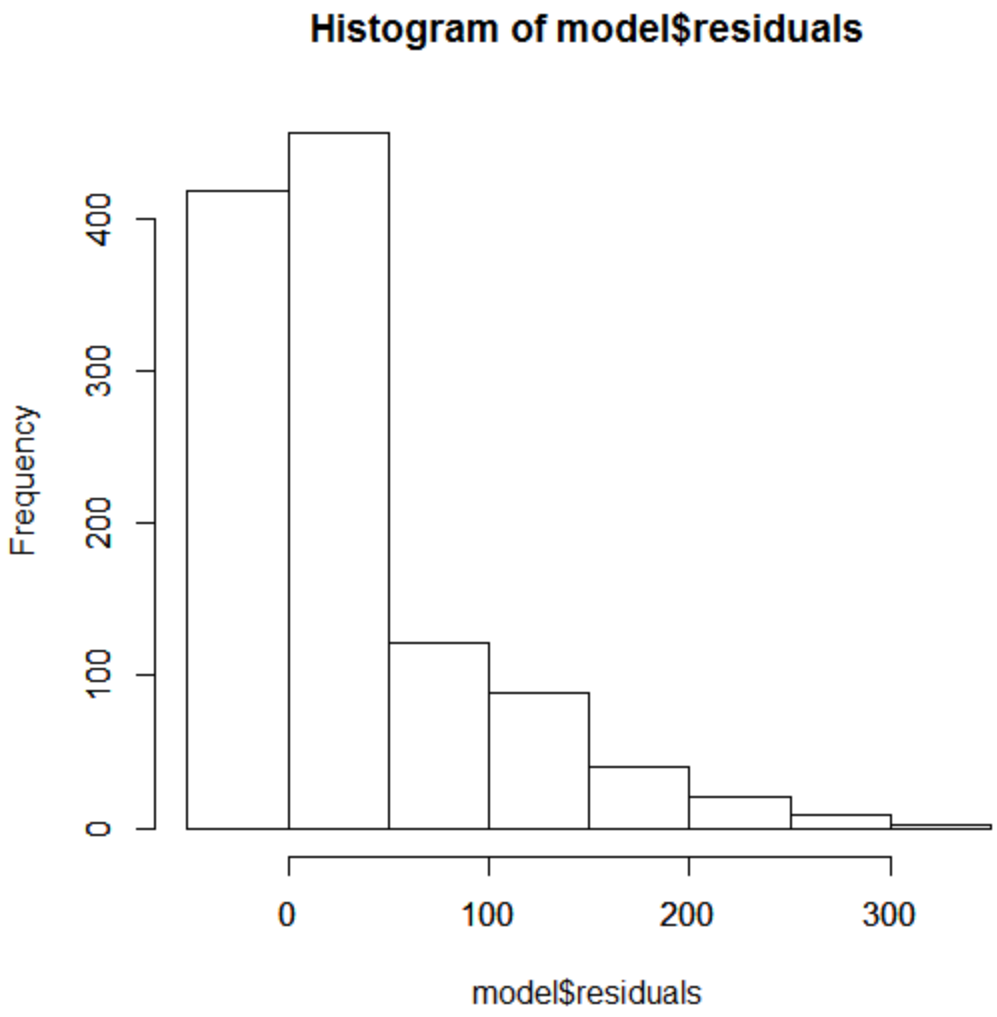
Create a mixed-effects model using lme, and summarize:

```
model <- lme(weight~Diet, random= ~Time|Chick, data=ChickWeight)

summary(model)
```

```
> summary(model)
Linear mixed-effects model fit by REML
 Data: ChickWeight
       AIC      BIC    logLik
  4909.447 4944.268 -2446.724

Random effects:
 Formula: ~Time | Chick
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev    Corr
(Intercept) 31.017008 (Intr)
Time         9.204751 -0.997
Residual    12.785139

Fixed effects: weight ~ Diet
               Value Std.Error  DF  t-value p-value
(Intercept) 54.92402  1.389555 528 39.52633  0.0000
Diet2        2.85461  2.364553  46  1.20725  0.2335
Diet3        1.99684  2.364553  46  0.84449  0.4028
Diet4        9.27429  2.367623  46  3.91713  0.0003
 Correlation:
      (Intr) Diet2  Diet3
Diet2 -0.588
Diet3 -0.588  0.345
Diet4 -0.587  0.345  0.345

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.72764450 -0.56576747 -0.03692424  0.47655985  3.47296794

Number of Observations: 578
Number of Groups: 50
>
```

Model residuals

Residuals are not normally distributed:



This is about as far as I got with this question.

MPG with GAM – outlier removal

```
# Assaf's lecture code (adapted)
library(mgcv)
setwd("C:/Users/Rod/SkyDrive/R/301/Week05")
autos = read.csv("autoMPGtrain.csv", as.is = TRUE)
autos$continent = factor(autos$continent)
autos$name = tolower(autos$name)
autos$diesel = grepl("diesel", autos$name)
autos$diesel[autos$name == "mercedes-benz 240d"] = TRUE
autos$cylgroup = cut(autos$cyl, c(2, 5.5, 6.5, 9))
gam4=gam(mpg~continent+diesel+cylgroup+s(weight,by=cylgroup)+s(year,k=
13)+s(accel)+s(hp),data=autos,select=TRUE)
plot(gam4,pages=1)
```

```
# Assaf used an rmse function.  Implement it.
rmse <- function(predicted, observed) { err <- sqrt(mean((predicted - observed)^2)); err }

autest = read.csv("autoMPGtest.csv", as.is = TRUE)
autest$continent = factor(autest$continent)
autest$name = tolower(autest$name)
autest$diesel = grepl("diesel", autest$name)
autest$diesel[autest$name == "mercedes benz 300d"] = TRUE
autest$cylgroup = cut(autest$cyl, c(2, 5.5, 6.5, 9))
gampreds = predict(gam4, newdata = autest)
rmse(gampreds, autest$mpg)
```

[1] 2.440786

```
setwd("C:/Users/Rod/SkyDrive/R/301/Week05")
autos = read.csv("autoMPGtest.csv")
str(autos)
which(autos$mpg > 46)
```

```
> autos[which(autos$mpg > 46),]
      name cyl volume hp weight accel year continent  mpg
86 mazda glc   4     86 65   2110  17.9   80       3 46.6
```

Personal note: The outlier is the 86 Mazda GLC, which was sold the next year as the 323. I bought an 87 Mazda 323 for $8500, and it was a fantastic car. It was breathtakingly fast. It got rear-ended in an I-5 traffic jam when there were 666 miles ominously on the odometer. I got the car and my neck fixed and drove it for another 140,000. It only got about 35 mpg on the road, so I suspect that the 46 should have been a 36. Contrary to appearance, I am not a “car guy”, and currently drive a 17 year old Honda Civic with 256,000 miles on it.

Remove the outlier, which happens to be the only mileage greater than 46, and predict. Get the RMS error.

```
idx = which(autest$mpg < 46)
gampreds2 = predict(gam4, newdata = autest[idx,])
rmse(gampreds2, autest$mpg[idx])
[1] 2.198124
```

More robust comparisons

```
abse <- function(predicted, observed) { err <- abs(predicted - observed); sum(err) }
```

```
>
> # Get vectors of absolute errors and do a summary to compare 75%
percentile (3rd quartile).
> abserrs = abs(gampreds - autest$mpg)
> abserrs2 = abs(gampreds2 - autest$mpg[idx])
> summary(abserrs)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
 0.04705  0.40680  1.37800  1.77800  2.66400 10.84000
> summary(abserrs2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.04705 0.40670 1.36900 1.68600 2.65200 6.39300
>
> ( abse(gampreds,autest$mpg) )
[1] 177.7619
> ( abse(gampreds2, autest$mpg[idx])  )
[1] 166.9268
```

Comparison of metrics

Now we have three metrics to compare models:

- RMS Error
- Absolute Error
- Comparison of third quartile of absolute errors.

Are these metrics really more robust? Well, as always in statistics, that depends. If the emphasis is to have less noise, then the 3rd quartile comparison metric appears reduces the effect of the outlier for the purpose of comparison. However, if I want less uncertainty in my prediction, that metric would be deceptive, and a low value on either of the other two metrics would be more meaningful.

A New Metric – the Assafian (since Reynolds, Prandtl, and Mach shouldn't get to have all the fun)

```
Assafian <- function(predicted, observed) {
        # Get the difference in 3rd quartiles.
        summaryPredicted = summary(predicted)
        summaryObserved = summary(observed)
        label = '3rd Qu.'
        percentileDiff = abs(summaryPredicted[label] - summaryObserved[label])

        # Get the absolute difference between predicted and observed.
        absdiff <- sum(abs(predicted - observed));

        # Take the mean of the two results.
```

```
        assafian = mean(c(absdiff, percentileDiff))

        assafian
}

# Tested
> (Assafian(gampreds,autest$mpg))
[1] 89.02096
> (Assafian(gampreds2, autest$mpg[idx]))
[1] 83.56839
```

Try my best parsimonious model from the dreaded homework 4:

```
> gam5 = gam(mpg ~ weight + year + continent + diesel, data=autos, select=TRUE)
> gampreds5 = predict(gam5, newdata = autest)
> (Assafian(gampreds5, autest$mpg))
[1] 123.908  (Well, that went in the wrong direction…)

gam6 = gam(mpg ~ cyl + volume + hp + weight + accel + year + continent + diesel + cylgroup, data=autos,
select=TRUE)
> gampreds6 = predict(gam6, newdata=autest)
> (Assafian(gampreds6, autest$mpg))
[1] 121.3225 (headed in the right direction, but not fast enough)
```