

Project

Winston

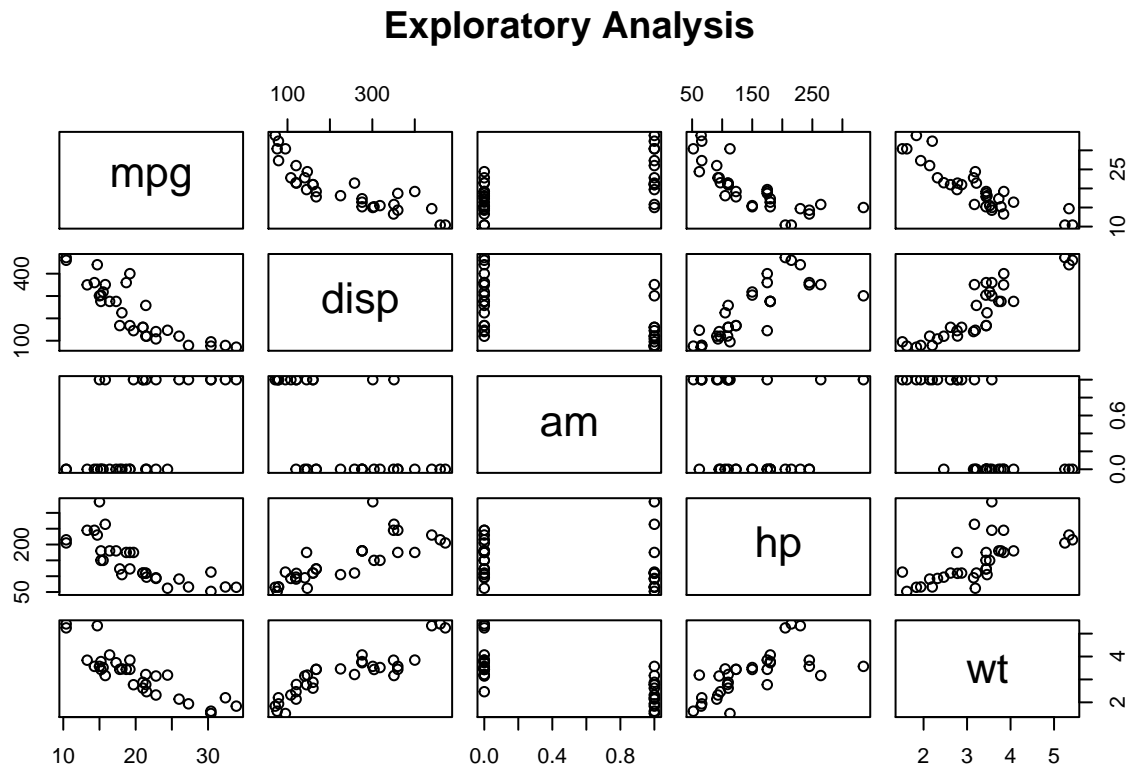
November 9, 2014

Executive Summary

exploratory data analysis

```
data(mtcars)

pairs(~mpg+disp+am+hp+wt,data=mtcars,
      main="Exploratory Analysis")
```



There are data for 32 different cars in the data set.

The data present are *mpg*, *cyl*, *disp*, *hp*, *drat*, *wt*, *qsec*, *vs*, *am*, *gear*, *carb*.

mpg Miles/(US) gallon *cyl* Number of cylinders *disp* Displacement (cu.in.) *hp* Gross horsepower *drat* Rear axle ratio *wt* Weight (lb/1000) *qsec* 1/4 mile time *vs* V/S *am* Transmission (0 = automatic, 1 = manual) *gear* Number of forward gears *carb* Number of carburetors

An exploratory analysis of the dependencies of *mpg* on several of the factors in the *mtcars* dataset show there are multiple dependencies. To understand how transmission type affects *mpg* we need account for these other factors.

As you can see from the graph, displacement and horsepower tend to follow weight. So we'll focus on correcting for the weight factor in *mpg* and look at residual.

linear model

The linear model shown in the graph above clearly does not fit the observed behavior.

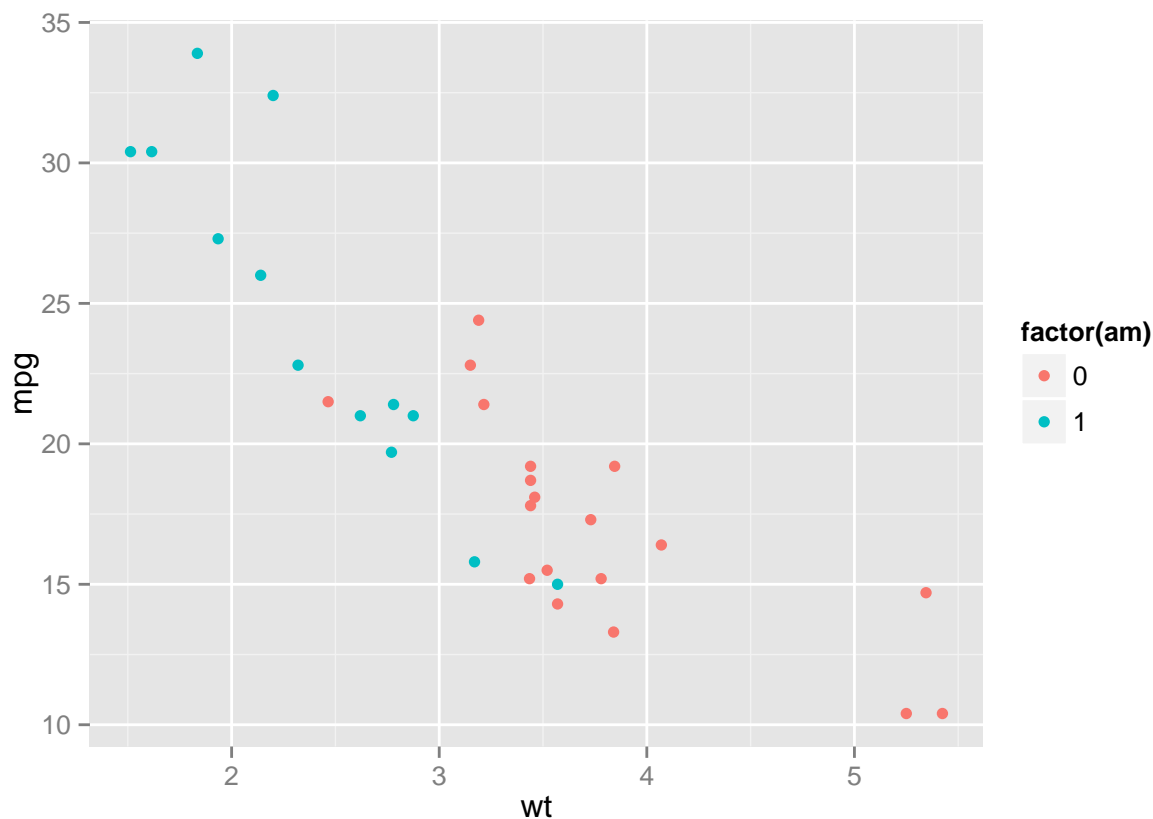
The reason the linear fit doesn't work well is because mpg is a kind of *reciprocal* energy (gallons of gasoline are equivalent to a measure of energy). The energy to accelerate a mass m to speed v is proportional to m so it makes sense to look at a reciprocal.

This suggests an inverse relationship between *mpg* and *weight* should be explored.

```
library(plyr)
library(ggplot2)

p<-ggplot(mtcars, aes(wt, mpg, color=factor(am))) + geom_point()
```

p



nonlinear model of mpg versus wt

To fit the data try a functional relationship $mpg \sim a + b/wt$ with a and b constants to be determined. The graph below shows the fit.

```
carnls <- nls(mpg~a1 + b1/hp, mtcars, start = list(a1=1, b1=50))
```

carnls

```
## Nonlinear regression model
```

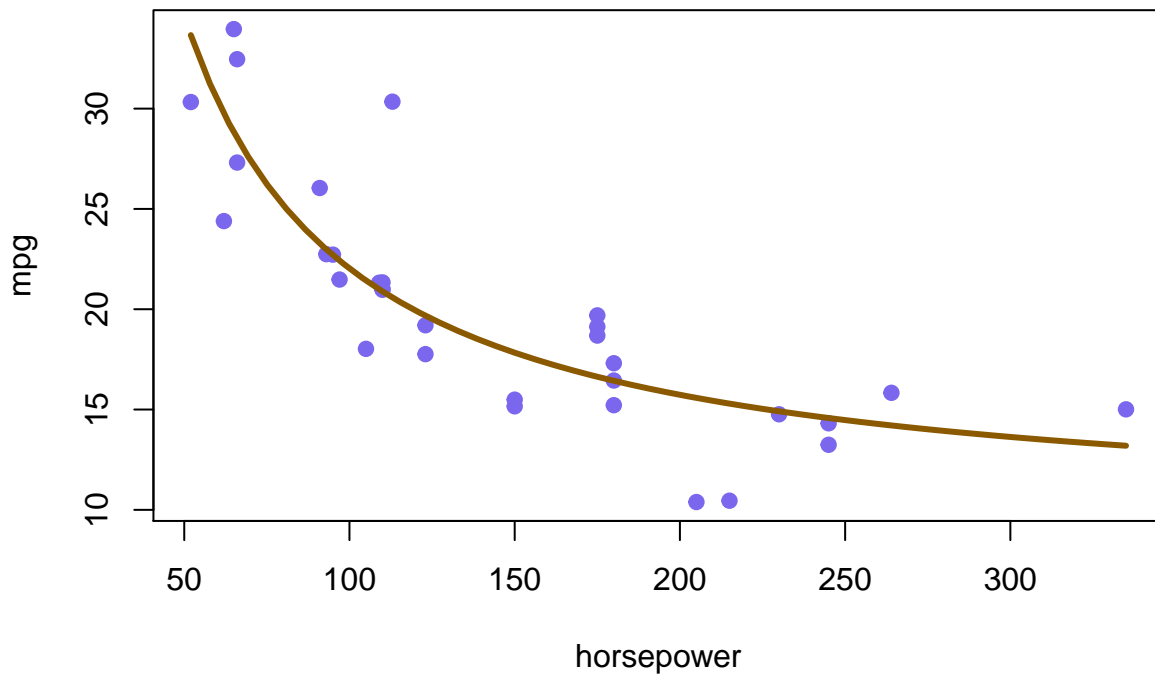
```
## model: mpg ~ a1 + b1/hp
## data: mtcars
## a1 b1
## 9.434 1259.881
## residual sum-of-squares: 294.9
##
## Number of iterations to convergence: 1
## Achieved convergence tolerance: 3.841e-08
```

```
ss<-seq(from = min(mtcars$hp), to = max(mtcars$hp), length = 50)
```

```
plot(jitter(mpg,4)~hp, data=mtcars, pch=19, col="slateblue2", xlab="horsepower", ylab="mpg")
```

```
yy <- predict(carnls,list(hp = ss))
```

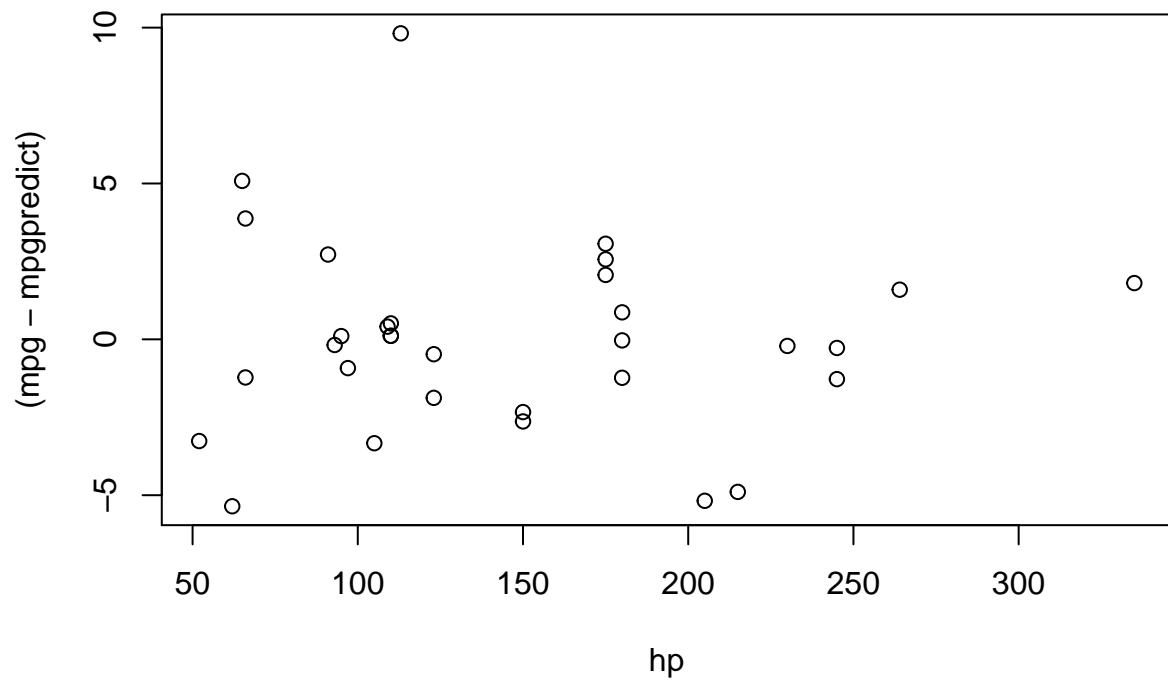
```
lines(ss, yy, col = "orange4", lwd=3)
```



Model Residuals

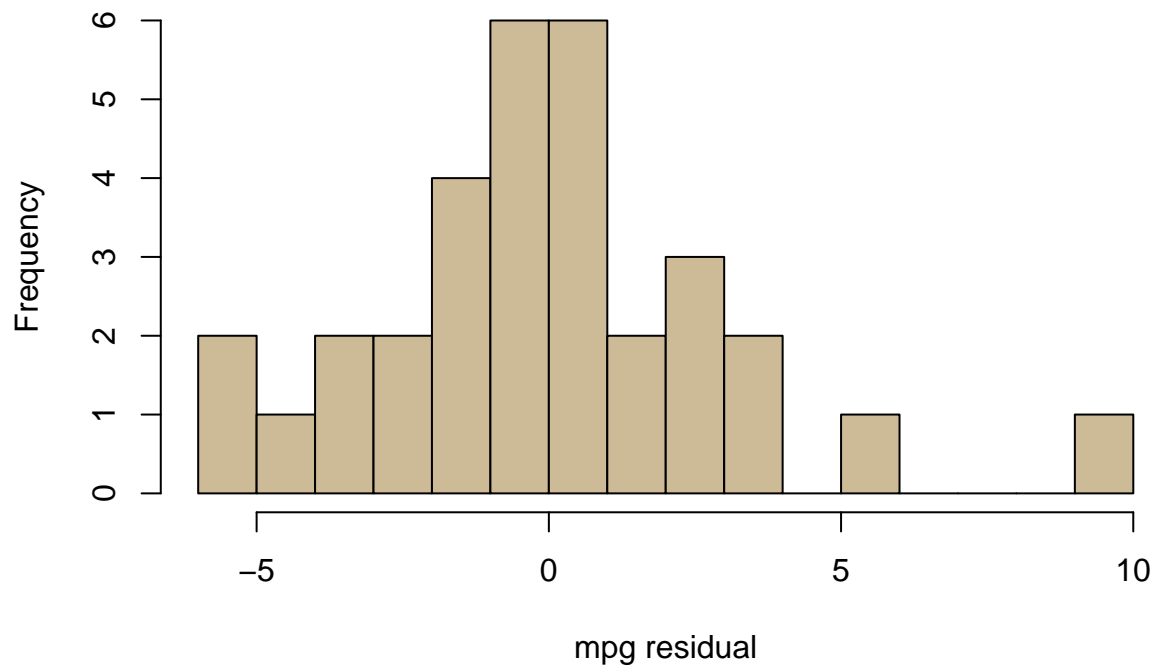
```
mpgpredict<-predict(carnls)
```

```
plot((mpg-mpgpredict)~hp, data=mtcars)
```



```
hist(mtcars$mpg-mpgpredict, breaks=18, col="wheat3", xlab="mpg residual", main="Histogram of mpg residuals")
```

Histogram of mpg residuals



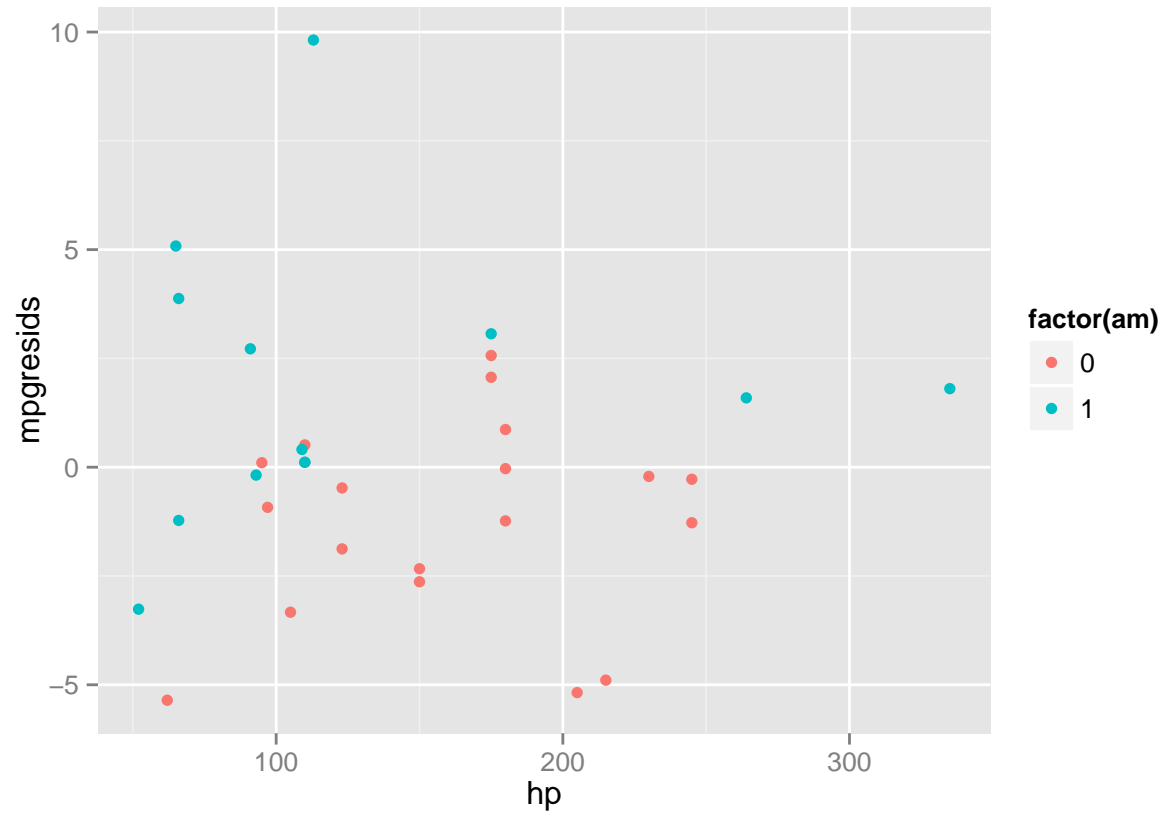
```
library(plyr)

mpgresids <- mtcars$mpg-mpgpredict

mtcars2<-cbind(mtcars, mpiresids)
```

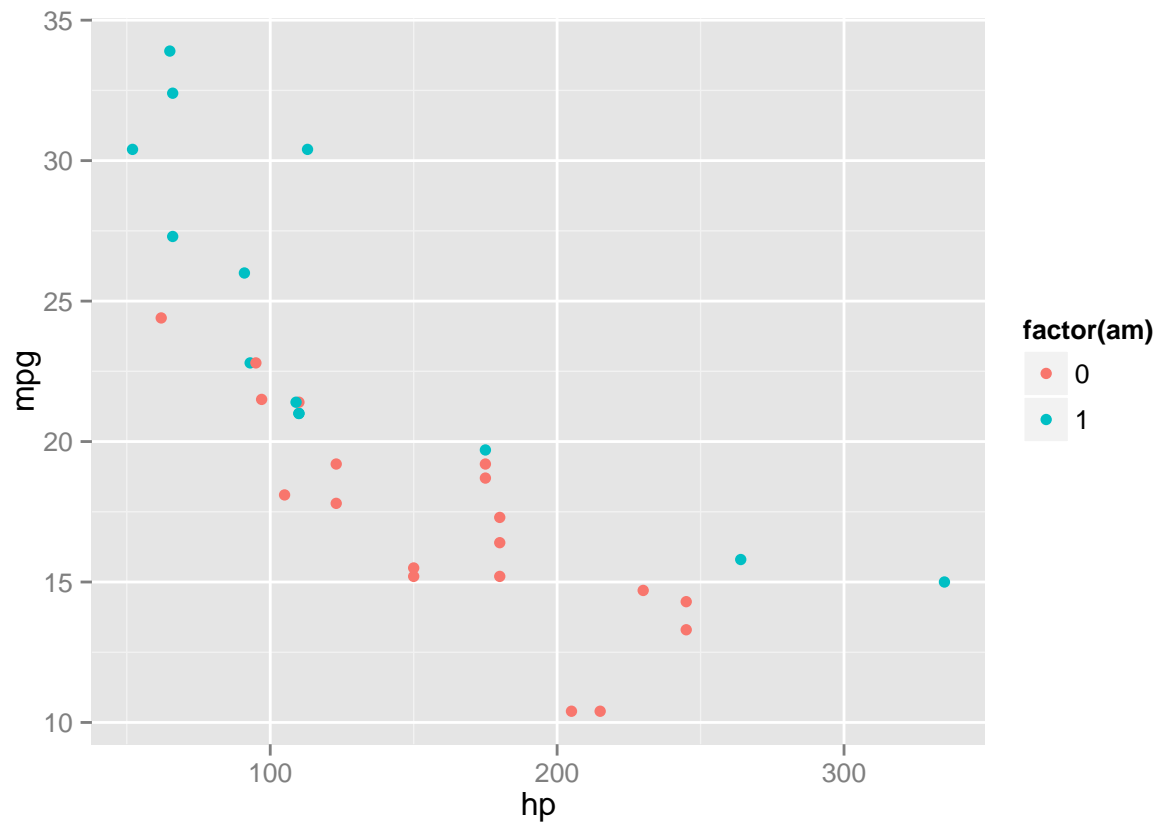
```
p<-ggplot(mtcars2, aes(hp, mpgresids, color=factor(am))) + geom_point()
```

p



```
p<-ggplot(mtcars, aes(hp, mpg, color=factor(am))) + geom_point()
```

p



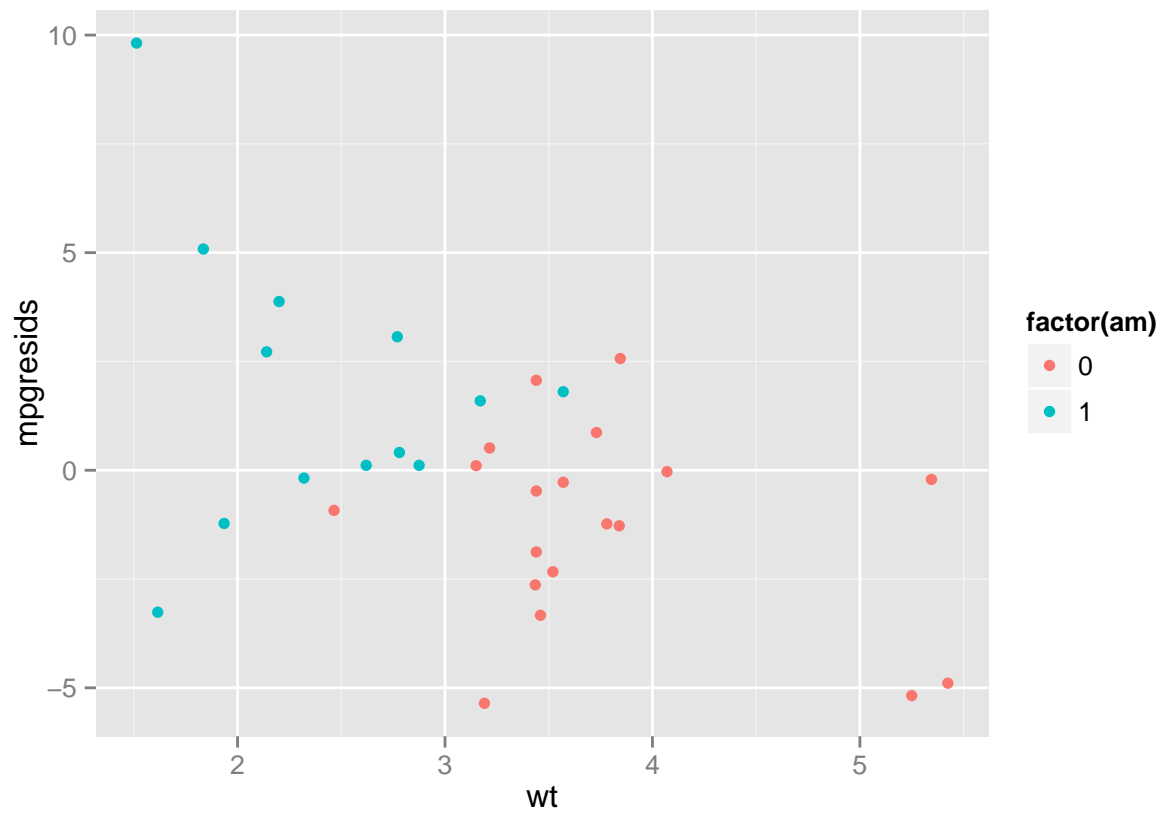
```
library(plyr)

mpgresids <- mtcars$mpg-mpgpredict

mtcars2<-cbind(mtcars, mpgresids)

p<-ggplot(mtcars2, aes(wt, mpgresids, color=factor(am))) + geom_point()

p
```



```
p<-ggplot(mtcars, aes(wt, mpg, color=factor(am))) + geom_point()
```

```
p
```

