Multicolinearity

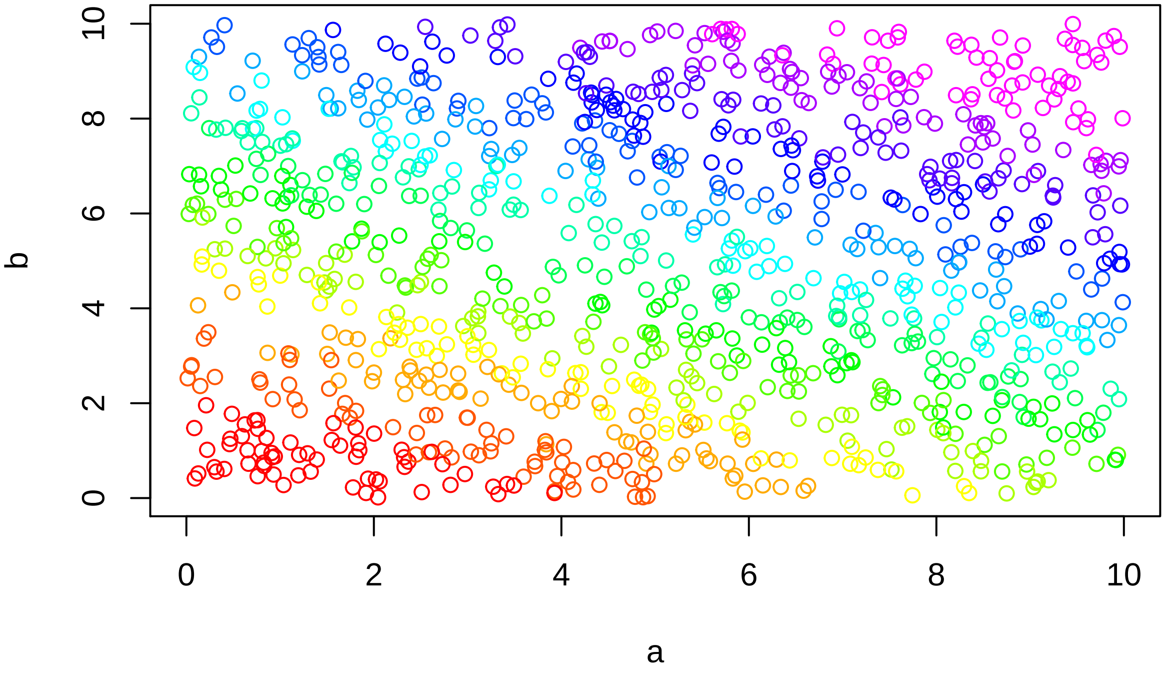
Bob Horton

March 18, 2015

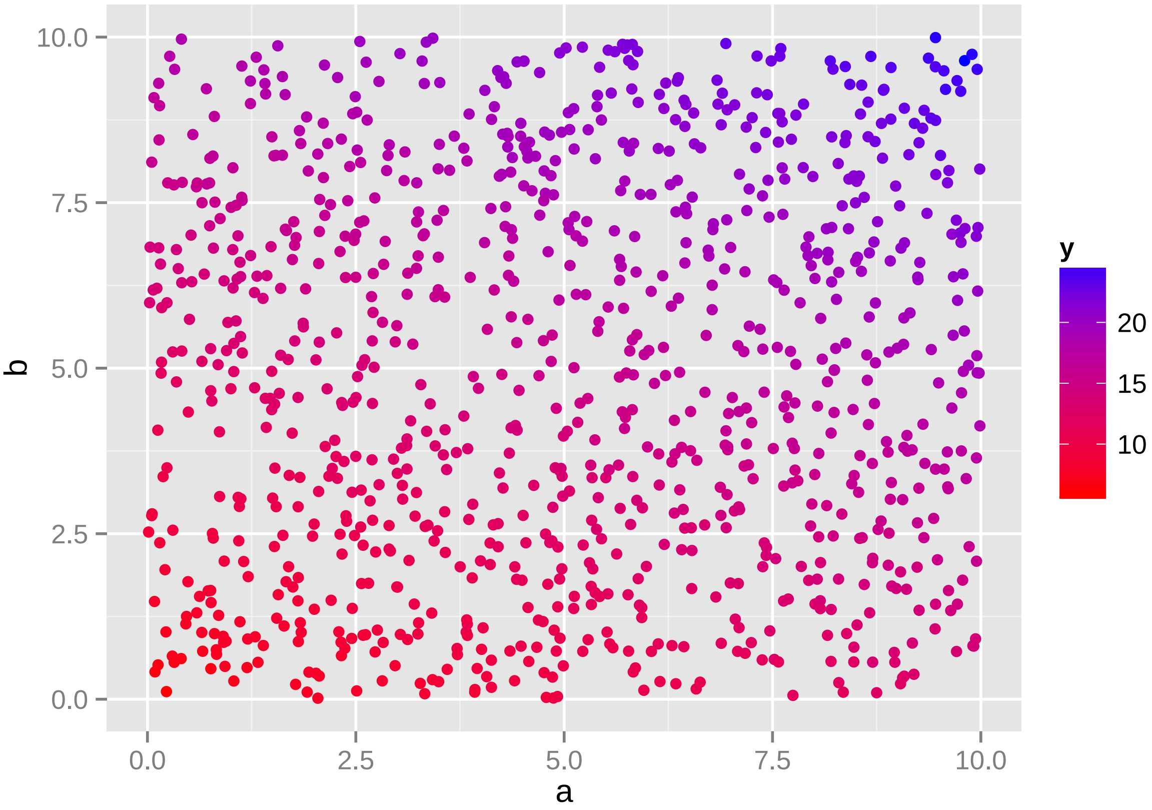
```
N <- 1e3    # 1e5

a <- runif(N, min=0, max=10)
b <- runif(N, min=0, max=10)

s <- 1.2

y <- 6 + 0.7 * a + 1.2 * b + rnorm(N, sd=0.2)

df1 <- data.frame(a, b, y)
```

To display three dimensions, we can use color. Here we break the y values into a series of ranges, and assign a color to each range. Colors are made by the `rainbow` function, which makes a series of hues spanning the spectrum.

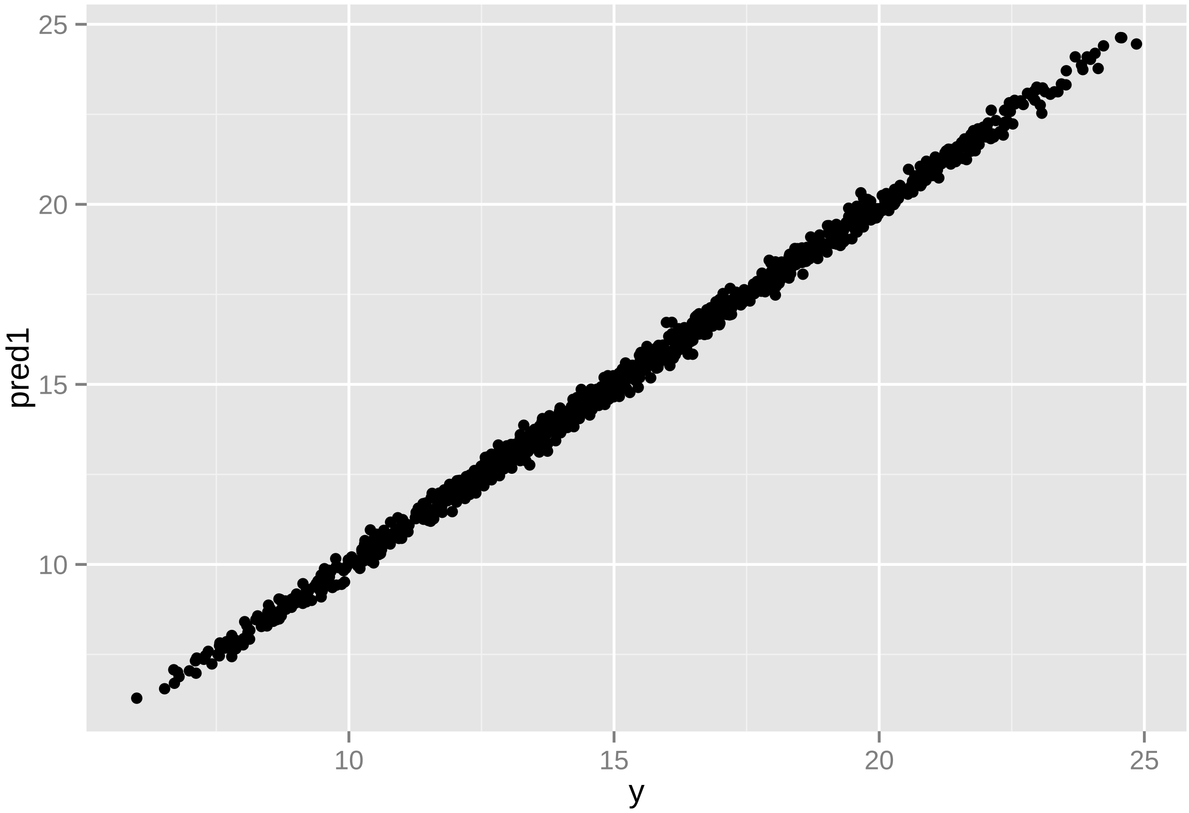
```
df1$y_bucket <- cut(y, breaks=quantile(y, probs=0:16/16))
rbow <- rainbow(16, end=5/6)

with(df1, plot(x=a, y=b, col=rbow[y_bucket]))
```



```
library(ggplot2)
ggplot(df1, aes(x=a, y=b, col=y)) +
    geom_point() +
    scale_colour_gradient(low="red", high="blue")
```



```
fit1 <- lm( y ~ a + b, data=df1)

df1$pred1 <- fit1$fitted

ggplot(df1, aes(x=y, y=pred1)) + geom_point()
```



```
summary(fit1)
```

```
##
## Call:
## lm(formula = y ~ a + b, data = df1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73273 -0.15768 -0.00722  0.15179  0.64761
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.987215   0.017352   345.0   <2e-16 ***
## a           0.701205   0.002347   298.8   <2e-16 ***
## b           1.201904   0.002321   517.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2135 on 997 degrees of freedom
## Multiple R-squared:  0.9973, Adjusted R-squared:  0.9973
## F-statistic: 1.866e+05 on 2 and 997 DF,  p-value: < 2.2e-16
```

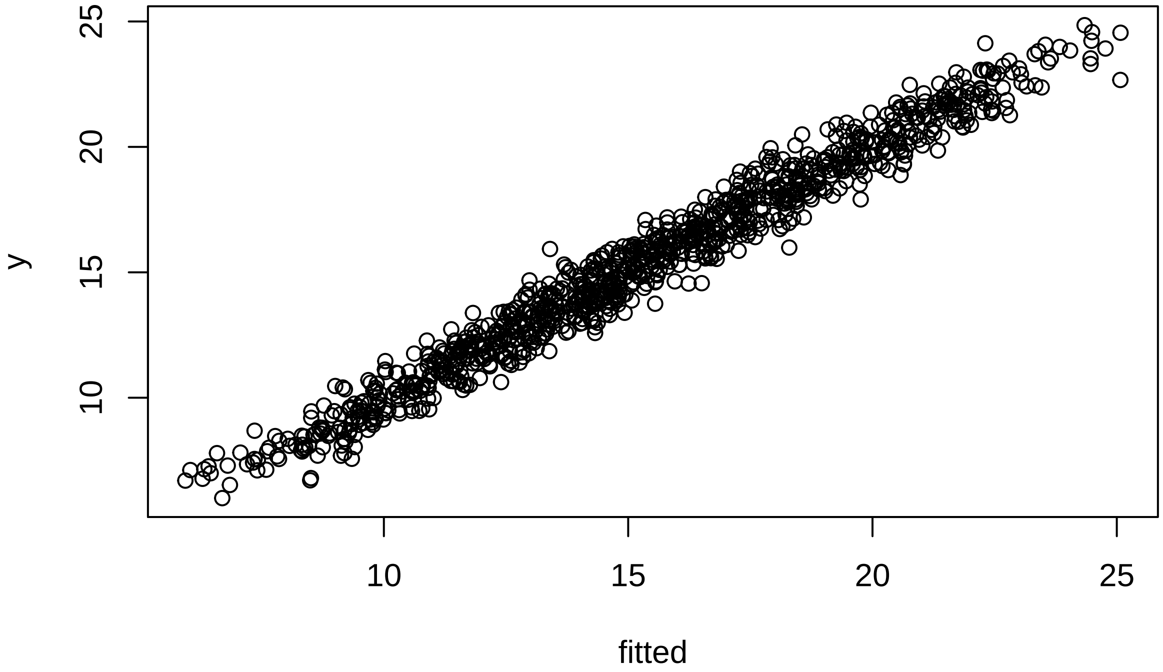
```
###
# (I want the fit to be good, but the coefficients to be non-significant.
# Try more multicollinear columns:
```

Matrix version

```r
num_a_cols <- 4
num_b_cols <- 4

X_signal <- matrix( c(rep(a, num_a_cols), rep(b, num_b_cols)), ncol=(num_a_cols + num_b_cols) )
X_noise <- matrix( rnorm( (num_a_cols + num_b_cols) * N), ncol=(num_a_cols + num_b_cols) )
X <- X_signal + X_noise
df2 <- cbind(data.frame(X), y)

fitA <- lm(y ~ ., data=df2)
summary(fitA)
```

```
##
## Call:
## lm(formula = y ~ ., data = df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4042 -0.4908 -0.0143  0.4939  2.5290
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.29979    0.05802 108.579  < 2e-16 ***
## X1           0.21362    0.02087  10.235  < 2e-16 ***
## X2           0.15282    0.01977   7.728 2.66e-14 ***
## X3           0.15720    0.02009   7.823 1.32e-14 ***
## X4           0.15132    0.02017   7.504 1.38e-13 ***
## X5           0.31577    0.01959  16.119  < 2e-16 ***
## X6           0.27937    0.02037  13.717  < 2e-16 ***
## X7           0.28543    0.01989  14.352  < 2e-16 ***
## X8           0.27701    0.02073  13.361  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7215 on 991 degrees of freedom
## Multiple R-squared:  0.9698, Adjusted R-squared:  0.9695
## F-statistic:  3973 on 8 and 991 DF,  p-value: < 2.2e-16
```

```r
df2$fitted <- fitA$fitted
plot(y ~ fitted, data=df2)
```



```
df2$fitted <- fitA$fitted
```