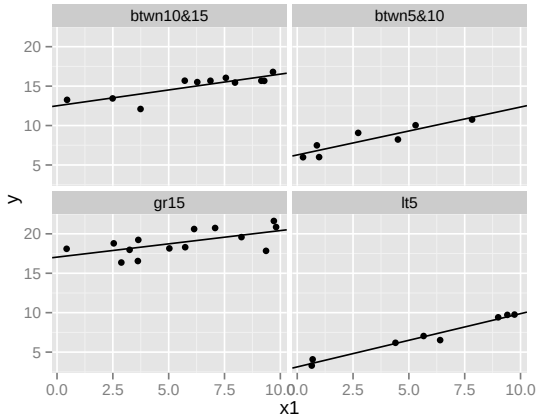
Stat 505 Assignment 7

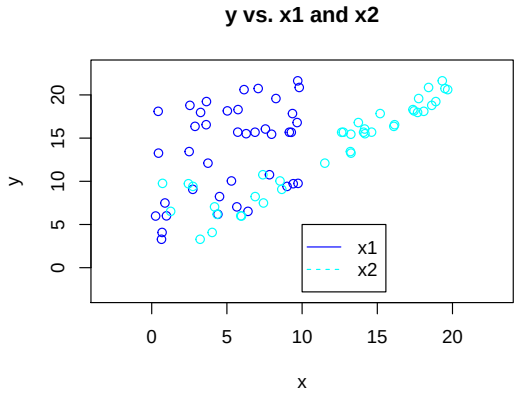
October 24, 2014

1. (a) Below is the summary for the interaction model. The model fits very well (adjusted $R^2 = 0.99$). There is strong evidence of a relationship between $x1$ and the mean of y when $x2$ is 0 (p-value$<$ 0.0001 from t-stat$=$ 16.43 on 36 df). A one unit increase in $x1$ is estimated to be associated with a 0.912 unit increase is the mean of y when $x2$ is held constant at 0, with a 95% confidence interval from a 0.799 to a 1.024 unit increase. There is also strong evidence of a relationship between $x2$ and the mean of y when $x1$ is zero (p-value$<$ 0.0001 from t-stat$=$ 34.190 on 36 df). There is also strong evidence that the relationship between $x1$ and y changes across $x2$ (p-value$<$ 0.0001 from t-stat$=$ -8.225 on 36 df). A one unit increase in $x1$ is estimated to be associated with a 0.037 unit decrease is the mean of y when $x2$ is held constant at 1, with a 95% confidence interval from a 0.0466 to a 0.0282 unit decrease.

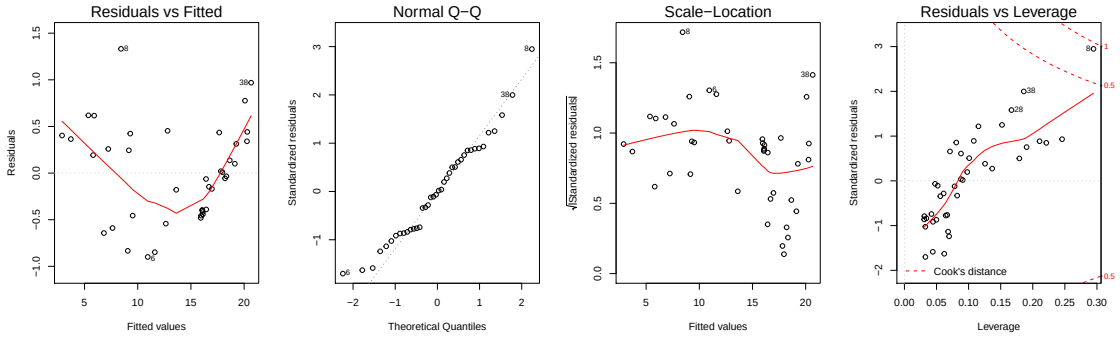
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.9170	0.3568	-2.57	0.0145
x1	0.9117	0.0555	16.43	0.0000
x2	1.0216	0.0299	34.19	0.0000
x1:x2	-0.0374	0.0045	-8.23	0.0000

(b) First, I looked at the relationship between $x1$ and y and $x2$ and y separately. But, I really want to look at the relationship between each variable and y while holding the other variable constant. To get a rough visual display of this, I grouped the $x2$ variable into four different intervals so that I could get an idea of the relationsip between $x1$ and y as $x2$ changes. The lines on the plots below show the relationship between $x1$ and y for each grouping of $x2$ values. We could also do this same thing to show the relationship between $x2$ and y as $x1$ changes.





(c) The residual plots are below. The normal Q-Q plots shows that the distribution of the residuals has shorter tails than we would expect from a normal distribution. I am not worried about the normality assumption because parametric tests are robust to departures from normality, especially with short-tailed distributions. There is no obvious trend in the scale-location plot, so I am not worried about the homogeneity of variance assumption. There is some indication of curvature in the residuals vs. fitted values plot. This indicates that the assumption of linearity may not be met and we should consider adding a curvature term to the model. Additionally, Cook's distance identifies an outlier. I ran the model without this point and the estimates and standard errors change only slightly and our results do not change.



(d) The following are the 95% prediction intervals for each of the 20 missing data points. The 95% prediction intervals in the table quantify how confident we are about these predictions. I feel good about these predictions. Our model appears to fit well ($R^2 = 0.9904$), and we are predicting responses for values of $x1$ and $x2$ within the range of data for which we do have observations.

	fit	lwr	upr
1	14.89	13.76	16.02
2	18.38	17.23	19.53
3	6.98	5.78	8.18
4	10.63	9.50	11.77
5	18.31	17.17	19.46
6	13.49	12.39	14.60
7	4.03	2.87	5.19
8	9.94	8.78	11.10
9	6.05	4.90	7.19
10	12.99	11.84	14.15
11	19.45	18.31	20.59
12	16.17	15.06	17.28
13	9.18	8.05	10.30
14	15.63	14.49	16.76
15	5.09	3.94	6.24
16	6.78	5.64	7.92
17	3.49	2.32	4.67
18	15.10	13.99	16.21
19	8.83	7.71	9.96
20	15.91	14.78	17.04

2. We solve $1.01^{\hat{\beta}_1} = 1.008$ and $\hat{\beta}_1 = 0.8008$. Then we solve $log(30000) = \hat{\beta}_0 + 0.8008 log(66)$ and $\hat{\beta}_0 = 6.954$. Our estimated regression line is then...

$$\mu(log(\widehat{earn})|log(ht)) = 6.954 + 0.8008 log(ht)$$

If approximately 95% of people fall within a factor of 1.1 of predicted values, then $e^{2\hat{\sigma}} \approx 1.1$. Our residual standard deviation, $\hat{\sigma}$ would be approximately $\frac{log(1.1)}{2} = 0.0477$.

3. $R^2 = 1 - \hat{\sigma}^2/s_y^2$, so we need to calculate s_y^2 from s_x^2. From our regression equation, we have,

$$log(earn)_i = \hat{\beta}_0 + \hat{\beta}_1 log(ht)_i + e_i$$

$$Var[\widehat{log(earn)}_i] = 0.8008^2 Var[\widehat{log}(ht)] + \widehat{Var}[e_i] = 0.8008^2 * 0.05^2 + 0.0477^2 = 0.00388$$

Then,

$$R^2 = 1 - .0477^2/0.00388 = 0.414$$

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.0293	0.0311	0.94	0.3466
var2	-0.0175	0.0316	-0.56	0.5784

4. (a) See the the model output below. No, the slope coefficient is not statistically significant (p-value= 0.5784). We would not expect it to be significant because there is truly no relationship between variable 1 and variable 2.

   (b) In 3 of 100 simulations, the slope coefficient was found to be statistically significant (when truly the slope is 0). This is consistent with what we would expect to see - we would expect to make a type I error about 5% of the time when using a t-statistic cutoff of 2.

```
## [1] 3
```

```
pyth.dat <- read.table("~/Documents/Stat505/Homework/HW7/pyth.dat", head=T)
```

```
pyth.dat1 <- pyth.dat[1:40,]
lm.fit <- lm(y~x1*x2, data=pyth.dat1)
require(xtable)
xtable(summary(lm.fit))
```

```
pyth.dat1$x2.f <- ifelse(pyth.dat1$x2<=5,"lt5", ifelse(pyth.dat1$x2>5 & pyth.dat1$x2<10,
x2.fac <- factor(pyth.dat1$x2.f)
lm.plot <- lm(y~x1*x2.fac,data=pyth.dat1)
require(ggplot2)
line.data <- data.frame(b = c(3.12158,6.27177,12.50536,17.04328), m=c(0.67539,0.60772,0.
qplot(x1,y,data=pyth.dat1, facets=~x2.f, geom="point") + geom_abline(aes(intercept = b,
```

```
plot(c(rep(-5,23)) ~ c(rep(-5,23)), ylim=c(-3,23), xlim=c(-3,23), ylab="y", xlab="x", ma
points(y~x1, type="p", ylim=c(0,20), col=4, data=pyth.dat)
points(y~x2, type="p", col=5, data=pyth.dat)
with(pyth.dat, legend(10,5, c("x1", "x2"), col=c(4,5), lty=c(1,2), text.col="black", mer
```

```
par(mfrow=c(1,4))
plot(lm.fit)
```

```
x.new <- data.frame(x1=pyth.dat$x1[41:60], x2=pyth.dat$x2[41:60])
predictions <- predict(lm.fit, x.new, interval="prediction", level=0.95)
require(xtable)
xtable(predictions)
```

```
set.seed(84)
var1<-rnorm(1000,0,1)
var2<-rnorm(1000,0,1)
reg <- lm(var1~var2)
require(xtable)
xtable(summary(reg))
```

```
set.seed(94)
require(arm)
z.scores <- rep(NA,100)
for (k in 1:100) {
  var1 <- rnorm(1000,0,1)
  var2 <- rnorm(1000,0,1)
  fit<-lm(var2~var1)
  z.scores[k] <- coef(fit)[2]/se.coef(fit)[2]
}
sum(ifelse(abs(z.scores)>2,1,0))
```