Key for (A Somewhat Shorter) Long Problem Set 5

The file "Anscombe.RData" contains three sets of x–y data; the objects for each set are identified as

set 1: x1 and y1

set 2: x2 and y2

set 3: x3 and y3

First, complete a linear regression analysis for each set of data using the linear model $y = \beta_0 + \beta_1 x$. Examine the summary data for each regression analysis and discuss the similarities and differences between the results. **Do not plot the data before completing this part of your work!**

Next, for each data set, prepare a plot of the data and overlay your regression model and comment on your results.

Finally, for any data set where the regression model is inappropriate, propose a better linear model; you may wish to consider how you might modify the data sets. When you are finished, you should have a suitable model for each data set.

Your answers are due in class on Monday, October 10th.

Answers

First, we need to generate linear models and summary reports for each data set; the code here will accomplish this.

```
load("Anscombe.RData")
model.set1 = lm(y1 ~ x1)
summary(model.set1)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92127 -0.45577 -0.04136  0.70941  1.83882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## x1            0.5001     0.1179   4.241  0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
```

```
model.set2 = lm(y2 ~ x2)
summary(model.set2)
```

```
##
## Call:
```

```
## lm(formula = y2 ~ x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9009 -0.7609  0.1291  0.9491  1.2691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## x2             0.500      0.118   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6662, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
```

```
model.set3 = lm(y3 ~ x3)
summary(model.set3)
```

```
##
## Call:
## lm(formula = y3 ~ x3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1586 -0.6146 -0.2303  0.1540  3.2411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x3            0.4997     0.1179   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
```

As we see in the table below, all three sets of data return essentially identical results for the predicted y-intercept, b_0, the predicted slope, b_1, the coefficient of determination, r^2, the standard deviation about the regression, s_r, and the ratio of $MS_{regression}$ to $MS_{residual}$, which yields F_{exp}.

data set	b_0	b_1	r^2	s_r	F_{exp}
1	3.0001	0.5001	0.6665	1.237	17.99
2	3.001	0.500	0.6662	1.237	17.97
3	3.0025	0.4997	0.6663	1.236	17.97

The one difference that we can see in the summary reports are for the residual errors, but what is included in the summary is not particularly informative. Given the results in these summary reports, we have little reason to believe that there is any difference in the quality of the three models; that is, this view of the data seems equally good or equally poor at explaining the data.

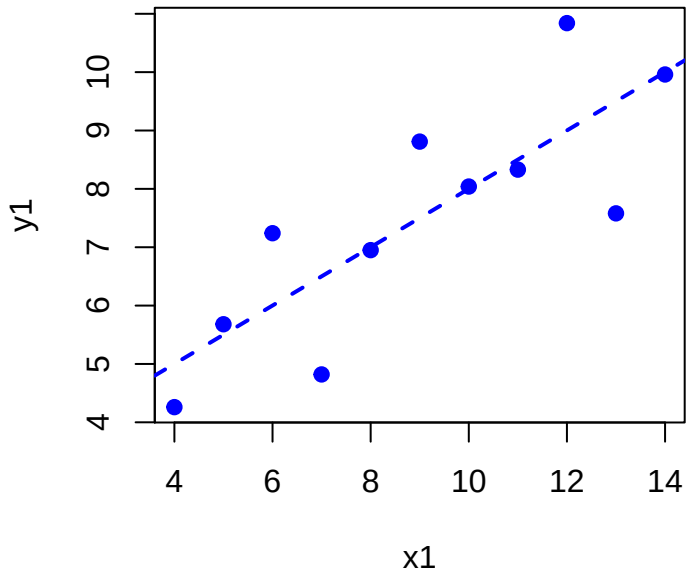

Figure 1: Regression results for Data Set 1

Next, for each data set, we prepare a plot of the data and overlay our regression model, with the results shown in Figures 1–3.

```
plot(x1, y1, pch = 19, col = "blue")
abline(model.set1, lwd = 2, col = "blue", lty = 2)
```

```
plot(x2, y2, pch = 19, col = "blue")
abline(model.set2, lwd = 2, col = "blue", lty = 2)
```

```
plot(x3, y3, pch = 19, col = "blue")
abline(model.set1, lwd = 2, col = "blue", lty = 2)
```

For the first data set a straight-line is not an unreasonable model; although the data themselves are subject to much uncertainty, there is nothing here to suggest that a different model will yield any improvement in the results.

For the second data set, it is clear that a straight-line is an inappropriate model and that a second-order polynomial equation likely will provide a better fit; let's try it out.

```
poly.set2 = lm(y2 ~ x2 + I(x2^2))
summary(poly.set2)
```

```
##
## Call:
## lm(formula = y2 ~ x2 + I(x2^2))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0013287 -0.0011888 -0.0006294  0.0008741  0.0023776
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.9957343  0.0043299   -1385   <2e-16 ***
## x2           2.7808392  0.0010401    2674   <2e-16 ***
## I(x2^2)     -0.1267133  0.0000571   -2219   <2e-16 ***
```

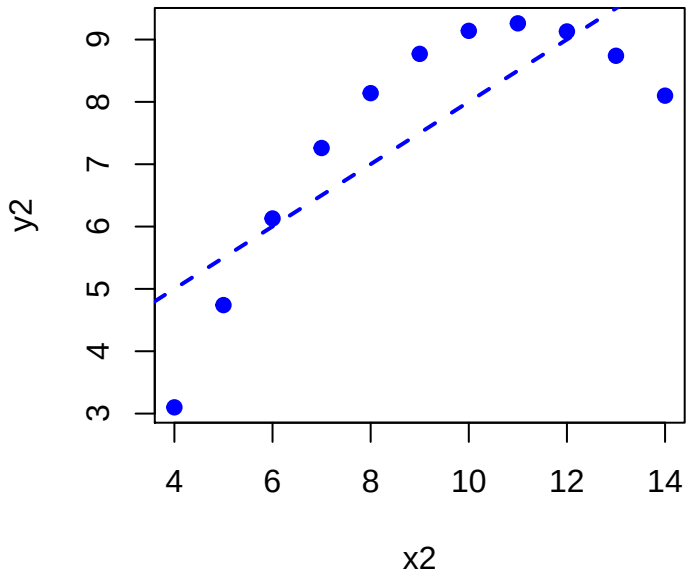

Figure 2: Regression results for Data Set 2

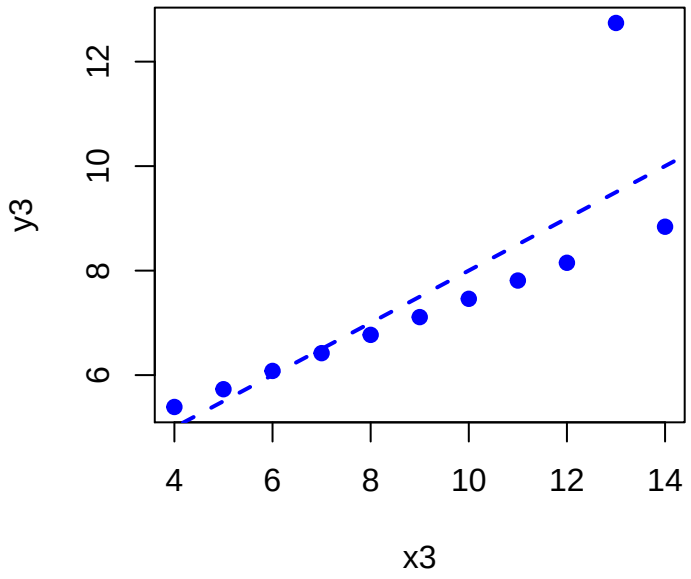

Figure 3: Regression results for Data Set 3

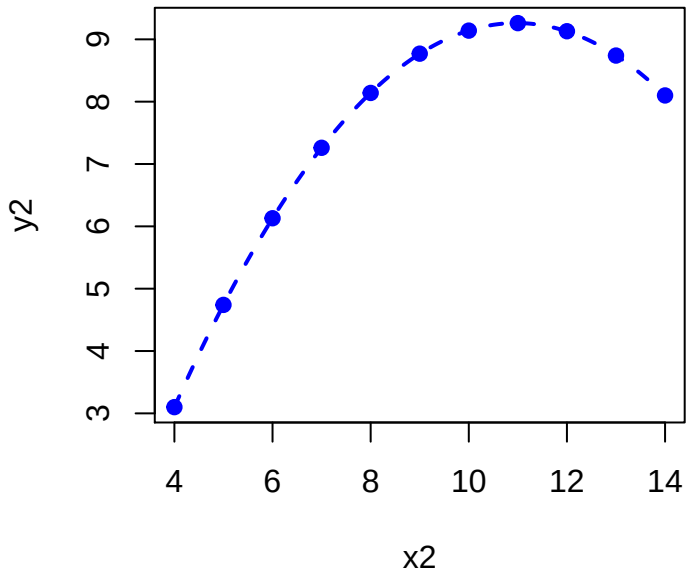

Figure 4: New regression results for Data Set 2

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.001672 on 8 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 7.378e+06 on 2 and 8 DF,  p-value: < 2.2e-16
```

```
plot(x2, y2, pch = 19, col = "blue")
xaxis = seq(4, 14, 0.01)
lines(x = xaxis,
      y = coef(poly.set2)[1] + coef(poly.set2)[2]*xaxis + coef(poly.set2)[3]*xaxis^2,
      lwd = 2, col = "blue", lty = 2)
```

For the third data set there is an apparent outlier with the remaining data seemingly consistent with our original first-order (straight-line) model. Let's remove the outlier, which has a value of 13 for x3, and then reanalyze the data.

```
# find the index entry for x3 that has a value of 13
which(x3 == 13)
```

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

```
# its the third entry, so let's remove it from both x3 and y3
x3 = x3[-3]
y3 = y3[-3]
new.set3 = lm(y3 ~ x3)
summary(new.set3)
```

```
##
## Call:
## lm(formula = y3 ~ x3)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0041558 -0.0022240  0.0000649  0.0018182  0.0050649
```

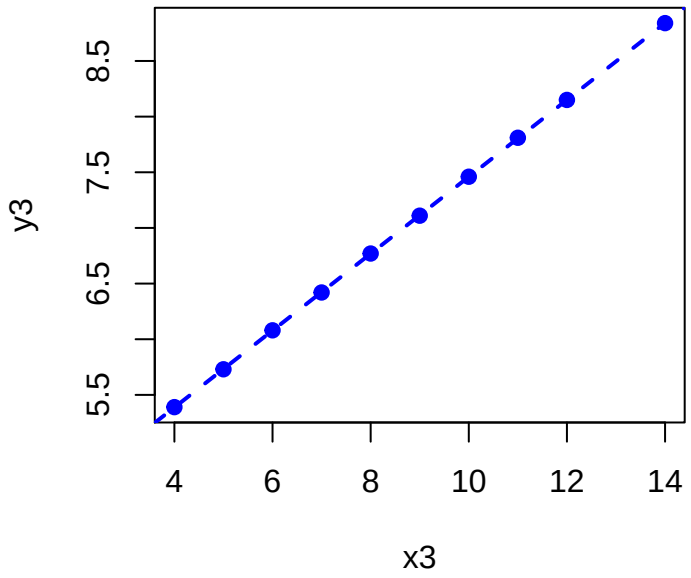

Figure 5: New regression results for Data Set 3

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.0056494  0.0029242    1370   <2e-16 ***
## x3          0.3453896  0.0003206    1077   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003082 on 8 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.161e+06 on 1 and 8 DF,  p-value: < 2.2e-16
```

```
plot(x3, y3, pch = 19, col = "blue")
abline(new.set3, col = "blue", lwd = 2, lty = 2)
```