

Regression Activity for Lab 2, ST512

1. Find the data entitled "Bigclass.txt" on the ST512 website.
2. Write a `data` step that will read these data in and create a data file. Note that `name` and `sex` are character variables. The syntax to establish this format is the dollar sign, which comes after the variable name in an `input` statement. The `cards` statement indicates that the data themselves are to be read in within the data step.

```
data kidz;
  input name $ age sex $ height weight;
  cards;
KATIE 12 F 59 95
LOUISE 12 F 61 123
JANE 12 F 55 74
...
run;
```

3. Plot weights against heights:

```
proc gplot data=kidz;
  plot height*weight;
run;
```

Use a different symbol or color for boys and girls. Also, change the awful default + symbol to a dot · using the `symbol SAS/GRAPH` statement, prior to `PROC GPLOT`.

```
symbol1 value=dot color=pink;
symbol2 value=dot color=blue;

proc gplot data=kidz;
  plot weight*height=sex;
run;
```

4. Would you say that taller children tend to weigh more? Is the association linear? Use `PROC CORR` to compute the sample correlation coefficient to quantify the linear association observed in the sample:

```
proc corr data=kidz;
  var weight height;
run;
```

Find the p -value for a test that the population correlation coefficient is 0.

5. Consider a linear regression of weight (y) on height (x). How much of the variance in weight of these children can be explained by heights, using the linear model?
6. Confirm the above by finding the appropriate statistic in the output from the REG procedure:

```
proc reg data=kidz;
    model weight=height;
run;
```

7. Consider estimating the mean weight among kids 5 feet tall ($x = 60$ inches). To do this you could
 - (a) use the missing y trick by adding an observation with `height= 60` and `weight= .`. Using an `output` statement, create a new dataset with fitted values, then ignore non-missing output.

```
data missy;
    weight=60; height=.;
run;
data kidz;
    set kidz missy;
run;
proc reg data=kidz;
    model weight=height;
    output out=predz p=p;
run;
proc print data=predz;
    where weight=.;
run;
```

- (b) Just look at Barbara, or some other 5-foot kid, in the dataset `predz`.
 - (c) Use an `estimate` statement in the GLM procedure:

```
proc glm data=kidz;
    model weight=height;
    estimate "mean weight for 5-foot kids" intercept 1 height 60;
run;
```

8. What is the estimated standard error associated with your guess at the mean weight among 5-footers?
9. What is your estimate of the standard deviation of the weights of all 5-footers?
10. Note the difference in the last two questions.

11. Produce a moderately fancy scatterplot by adding a `plot` statement within PROC REG:

```
proc reg data=kidz;
  model weight=height;
  plot weight*height;
run;
```

12. An activity using the R package.

- (a) Boot up R by double-clicking the icon.
- (b) Using the `read.table` command create an object in R that contains the data.
- Save the data “Bigclass.txt” as a textfile somewhere.
 - Using the menu in R, Change the directory to this same location (Click File, then choose “Change dir . . .”) then browse to the right place.)

```
> bigclass.dat <- read.table("Bigclass.txt",header=T)
```

- (c) Look at the first three rows of the object you created

```
> bigclass.dat[1:3,]
```

- (d) The heights and weights are in columns 4 and 5. Plot them.

```
> plot(bigclass.dat[,4:5])
```

- (e) Obtain the least squares regression line using the `lm` function, and store as an object. Then print the object to the screen, in particular, print the coefficients.

```
> bigclass.lsr <- lm(weight ~ height, data=bigclass.dat)
> bigclass.lsr
> coef(bigclass.lsr)
```

- (f) Overlay the regression line on the existing plot, using the `abline` command.

```
> abline(coef(bigclass.lsr))
```

13. Another activity with R: produce a plot depicting the decomposition of residuals into components due to regression and due to error using the script entitled “deviationplot-slr.r” (This plots will be like the ones in the file “corn-3plot.pdf”).

- (a) Examine script then save it in the current working directory

- (b) `source` the script.

```
> source("deviationplot-slr.r")
```

- (c) Use the function created by the script.

```
> deviations.slr.plot(bigclass.dat[,c(5,4)],"kid weights")
```