Statistics 411/511
Lab 8

Lab Instructions: If you want to work along with the TA in lab, please sit near the front. If you prefer to go to lab but work at your own pace, please sit near the back, and wait for the appropriate time to ask any questions.

Lab 8: Introduction to Simple Linear Regression

Objectives for this Lab

- Estimate simple linear regression parameters β_0 and β_1.
- Graph a scatterplot.
- Add a fitted line, a confidence band, and a prediction band to the scatterplot.
- Produce a residual plot.

1. Start up RStudio. Load the Sleuth3 package. If you are working in Bexell, you'll have to install the packages again as described in item 5(a) of Lab 1.

   ```
   > library(Sleuth3)
   ```

2. View the first few lines of the galaxy data, and create a scatterplot.

   ```
   > head(case0701)
   > qplot(Distance,Velocity,data=case0701)
   ```

   Note that this `qplot()` command looks very similar to the one we have used to produce side-by-side boxplots. The difference here is the lack of the `geom="boxplot"` and that `Velocity` is a numeric variable, not a categorical or "factor" variable. You can check this using `is.numeric()`.

   ```
   > is.numeric(case0701$Velocity)
   ```

3. Perform the regression, that is, fit a line to the points in the scatterplot. As we did when performing a one-way ANOVA, we'll save an "object" called `case0701.lm` that contains the analysis.

   ```
   > case0701.lm<-lm(Distance~Velocity,data=case0701)
   ```

   The function `lm()` is useful for any kind of "linear model" analysis, of which simple linear regression and one-way ANOVA are two examples. If you take ST 412/512, you will spend a lot of time with `lm()`.

4. Get the parameter estimates and their standard errors.

   ```
   > summary(case0701.lm)
   ```

   Compare your R output to the table of parameter estimates in Display 7.9. The row labeled "Constant" in Display 7.9 is labeled "(Intercept)" in the R output. This row refers to the parameter β_0 which is the y-intercept in the linear equation

   $$y = \beta_0 + \beta_1 x. \tag{1}$$

   Equation (1) states that y is a linear function of x. Compare this to the model formula on page 181 of the *Sleuth*, which says that the population mean of Y is a linear function of X.

   Find the point estimates and standard errors for β_0 and β_1 in the R output.

5. The R output from item 4 contains point estimates and standard errors for intercept β_0 and slope β_1. R keeps these values in the `lm` object `case0701.lm`.In particular, R calls the point estimates "coefficients":

   ```
   > case0701.lm$coefficients
   ```

   We can use these estimated coefficients to draw the estimated regression line on the scatterplot by adding a layer via `geom_abline()` to the `qplot()` command. `geom_abline()` needs two arguments, the intercept and slope.

   ```
   > qplot(Velocity,Distance,data=case0701) + geom_abline(slope=0.0014,intercept=0.3992)
   ```

   Your scatterplot should now have a beautiful regression line.

6. The `ggplot()` function is capable of drawing the fitted regression line automatically. Here's the command.

   ```
   > ggplot(case0701, aes(x=Velocity, y=Distance)) +
   +   geom_point() +
   +   geom_smooth(method=lm,se=FALSE)
   ```

   Notes: First, there are four `"+"` symbols on the above code. The two on the far left of the second and third lines are continuation prompts and are not in the code in Lab8.r. The two `"+"` symbols on the right of lines one and two are part of the command. They are telling `ggplot()` to *add layers*.

   The code `method=lm` tells R to fit a regression line. The code `se=FALSE` tells R *not* to include a confidence band.

7. Omitting `se=FALSE` will produce a plot with a 95% confidence band on the scatterplot. These are the curves labeled "95% confidence band for estimated means" on the figure in Display 7.11.

8. Add the "95% prediction band for an unknown distance" in Display 7.11. This we'll have to do by hand–`ggplot()` doesn't have a way to add these automatically.

   (a) Create predictions with lower and upper 95% prediction limits using `predict()`.

   ```
   > pred.intervals<-data.frame(predict(case0701.lm,interval="prediction"))
   ```

   R gives a warning message to remind you that prediction intervals are different from confidence intervals. We will discuss prediction vs. estimation in lecture (and see Section 7.4.3 of the textbook for details). If instead of `interval="prediction"`, you write `interval="confidence"`, you'll get confidence intervals.

   We put the `predict()` command within the `data.frame()` command to make `pred.intervals` a data frame. Otherwise it's a "matrix," and we can't refer to its columns by name.

   Speaking of column names, what are they?

   ```
   > head(pred.intervals)
   ```

   The lower and upper 95% prediction limits are called "lwr" and "upr." The first column contains the point predictions, the points on the estimated line corresponding to the Velocity values given in the original data. You can find these numbers in the "fits" column in Display 7.8.

(b) Join the output of `predict()` to the original data frame. This puts the data and the prediction bounds in the same data frame which we can then give to `ggplot()`.

```
> case0701.df2 <- cbind(case0701, pred.intervals)
> head(case0701.df2)
```

(c) Create the plot.

```
> ggplot(case0701.df2, aes(x=Velocity, y=Distance)) +
+   geom_point() +
+   geom_line(aes(y=lwr), color = "blue", linetype = "dashed") +
+   geom_line(aes(y=upr), color = "blue", linetype = "dashed") +
+   geom_smooth(method=lm)
```

This is the same code as in item 6 with two extra layers for the two dotted lines.

9. When we did the regression back in item 3, R calculated fitted values and residuals.

(a) Compare the first few of R's residuals to those in Display 7.8.

```
> head(case0701.lm$residuals)
```

The discrepancies are due to rounding error in Display 7.8.

(b) When we discuss assumptions for regression in Chapter 8, we'll see that a plot of the residuals vs. fitted values is a useful diagnostic, just as it was for one-way ANOVA. You can get the plot in the same way that we did before.

```
> plot(case0701.lm,which=1)
```

In regression, the "Fitted values" are obtained by plugging the observed explanatory variables (here Velocity) into the estimated regression equation. These are the point predictions of item 8(a).