Homework2

Ethan Ashby

2/10/2020

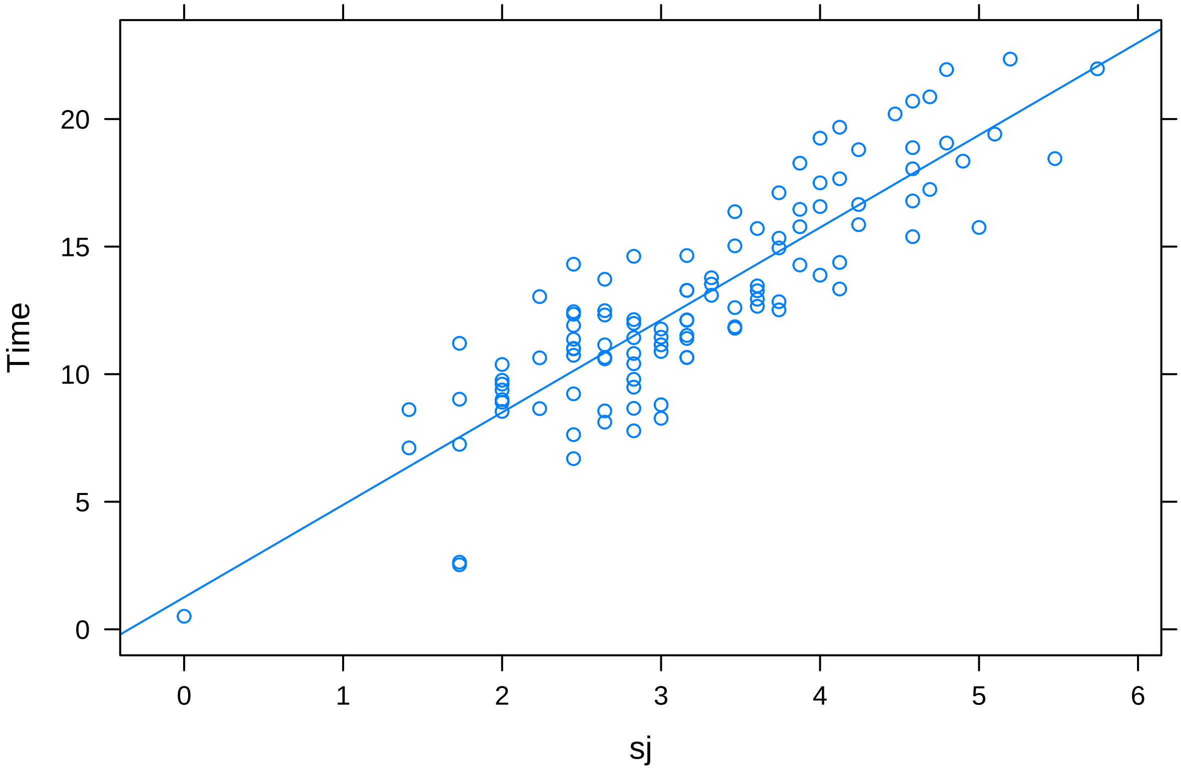
1. Explore lab 1 data

(a) Write down model that you think works.

```
labdata<-read.csv("Lab1data.txt")
summary(labdata)
```

```
##       Time            Jobs      
##  Min.   : 0.51   Min.   : 0.00  
##  1st Qu.:10.62   1st Qu.: 6.00  
##  Median :12.45   Median :10.00  
##  Mean   :12.93   Mean   :11.37  
##  3rd Qu.:15.73   3rd Qu.:15.00  
##  Max.   :22.35   Max.   :33.00  
```

```
labdata$sj <- with(labdata, sqrt(Jobs))
xyplot(Time~sj, data=labdata, type=c('r', 'p'))
```



```
#the sqrt transformation on 'Jobs' improves the linear fit! I feel comfortable applying the model here
```

```
#sqrt transform jobs data
labdata$sj <- with(labdata, sqrt(Jobs))

#fit lm to this data, with y intercept set to 0, since 0 jobs should take 0 minutes
fit<-lm(Time~0+sj, labdata)
```

```r
summary(fit)
```

```
##
## Call:
## lm(formula = Time ~ 0 + sj, data = labdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3618 -1.2946  0.1757  1.6378  4.5635
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## sj  3.97900    0.05674   70.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.016 on 110 degrees of freedom
## Multiple R-squared:  0.9781, Adjusted R-squared:  0.9779
## F-statistic:  4918 on 1 and 110 DF,  p-value: < 2.2e-16
```

$$time = 3.979 * \sqrt{jobs} \tag{1}$$

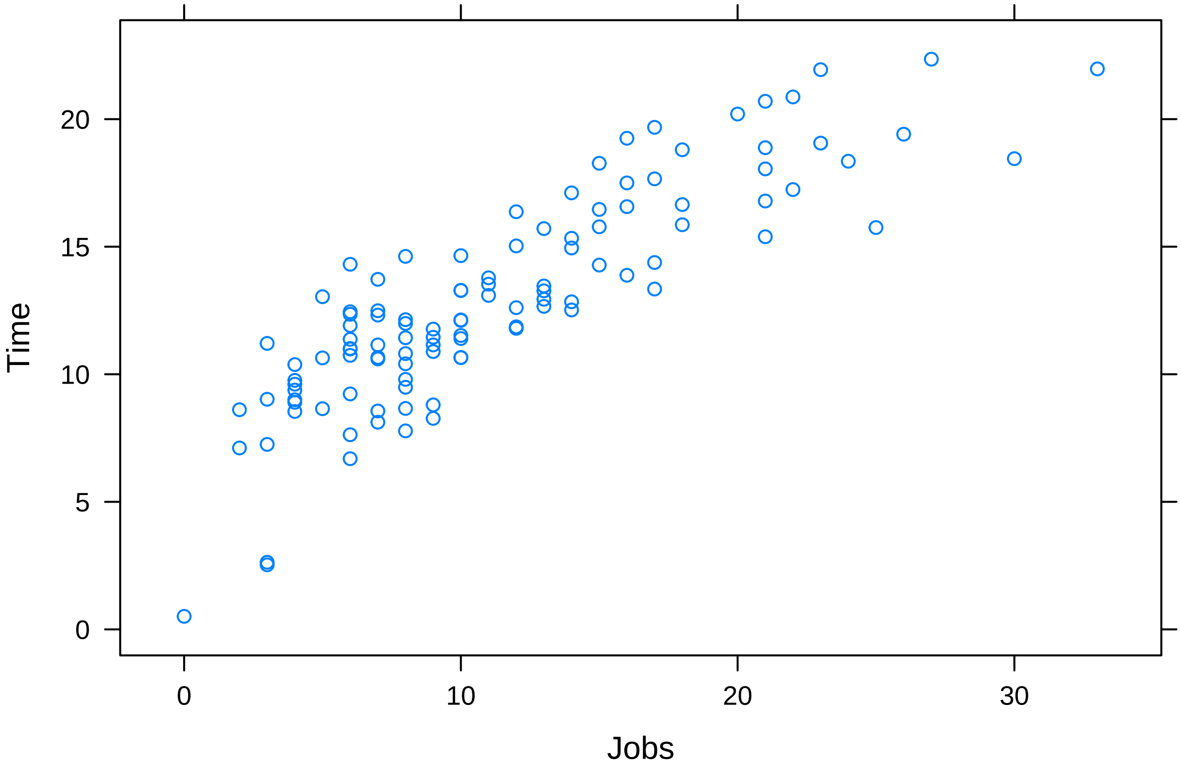
(b) Predict time required to do 10 jobs ($x^* = 10$)

```r
#Time required to do 10 jobs given our model
predict(fit, data.frame(sj=sqrt(10)))
```

```
##        1
## 12.58269
```

(c) In the context of this data, make sense of the form the regression relationship might take. In particular, discuss why we might see a linear relationship, a convex relationship, or the concave relationship seen in the actual data. Keep this specific to the variables we are studying.

```r
#relationship in the actual data
xyplot(Time~Jobs, labdata)
```



We might anticipate seeing a concave relationship (which we do see in the data) because a small number of jobs requires a mininum amount of time to complete. However, when posed with a large number of jobs, the process becomes more efficient and it takes less time to complete each job, resulting in a tapering off of the data at higher number of jobs. This accounts for the concavity of the data.

If the relationship is convex, then as the number of jobs increases, the process becomes less efficient and the time/job increases. So at a high number of jobs, the time required to complete explodes.

If the relationship is linear, then as the time/job is constant regardless of how many jobs we have.

2. Based on the work we did understanding the forms of the variances of the of the regression coefficients, discuss (confidence) interval estimation for a quantity like the one in 1(b). Should the size of the interval be a function of x^*? If so, how and why? Where do you think the smallest interval would be for a fixed confidence level?

The size of the interval should be a function of x^*. The formula for the variance of $\hat{\beta}_0$ is as follows:

$$s^2_{\hat{\beta}_0} = MSE[\frac{1}{n} + \frac{\bar{x}^2}{\sum(x_i - \bar{x})^2}]$$

Note that this formula asserts that the estimate of $\hat{\beta}_0$ is dependent on $\bar{x}$ and x_i among other things. To reduce the variability of our estimate for $\hat{\beta}_0$, the formula notes that the data should be gathered with mean $\bar{x} = 0$; thus, the smallest confidence interval is around $\bar{x}$. This suggests that proximity to the $\bar{x}$ is a key determinant of the variability of an estimate generated by a linear model. Thus, estimates like 1(b) are a function of x^*, since as x^* changes, the proximity to $\bar{x}$ changes. And from the regression coefficient formula, $\bar{x}$ is when the variability of our estimate is the lowest (and our confidence interval is smallest).

To wit, consider the example using our model: fit

```
#use predict.lm with argument 'interval="confidence"' to generate confidence interval for mean time at
summary(fit)
```

```
##
```

```
## Call:
## lm(formula = Time ~ 0 + sj, data = labdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3618 -1.2946  0.1757  1.6378  4.5635
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## sj  3.97900    0.05674   70.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.016 on 110 degrees of freedom
## Multiple R-squared:  0.9781, Adjusted R-squared:  0.9779
## F-statistic:  4918 on 1 and 110 DF,  p-value: < 2.2e-16
```

```
#Time to complete 10 jobs
predict.lm(fit, data.frame(sj=sqrt(10)), interval="confidence")
```

```
##        fit     lwr      upr
## 1 12.58269 12.2271 12.93828
```

```
#Time to complete 100000 jobs
predict.lm(fit, data.frame(sj=sqrt(100000)), interval="confidence")
```

```
##        fit     lwr      upr
## 1 1258.269 1222.71 1293.828
```

As evidenced by comparing confidence interval predictions for 10 and 100,000 jobs, changing the number of jobs affects the confidence interval width.

3. Show that the regression line necessarily passes through the point $(\bar{x}, \bar{y})$.

We know that this is true because least squares regression defines

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

. A rearrangement of this expression shows that

$$\hat{\beta}_0 + \hat{\beta}_1 \bar{x} = \bar{y}$$

, which demonstrates that $(\bar{x}, \bar{y})$ satisfy the equation of the line of best fit. Thus the line must pass through this point.

4. Consider a situation where the x variables are measured with error, so that rather than observing (xi, yi) we observe $(x_i + \eta_i, y_i)$ where the y_i are the responses at the actual x_i that we don't get to see and η_i is some mean 0 random variable (similar to ϵ_i). The consequence of this is that the least squares estimate of the slope will be biased. Using intuition, argue which direction the bias will be (negative bias: $\hat{\beta}_1$ will tend to be too small, vs positive bias).

Recall that

$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x})(y_i)}{\sum (x_i - \bar{x})^2}$$

Jittering the x values slightly in your measurement will not change the value of $\bar{x}$, since the η_i is normally distributed with mean=0. So some x_i will be positively perturbed, and some will be negatively perturbed, so the $\bar{x}$ will remain the same. Jittering the x values also won't change the value of $\sum (x_i - \bar{x})(y_i)$, since $\sum ((x_i + \eta_i) - \bar{x})(y_i) = \sum (x_i - \bar{x})(y_i) + \sum (\eta_i)(y_i)$ and $\sum (\eta_i)(y_i) = 0$ because the jitter is independent of the value of the response variable. $\sum ((x_i + \eta_i) - \bar{x})(y_i) = \sum (x_i - \bar{x})(y_i)$, so the jitter has no effect on the numerator of the $\hat{\beta}_1$ expression.

However, the denominator of the expression for $\hat{\beta}_1$ will increase. Compare the unperturbed sum ($\sum(x_i - \bar{x})^2$) and the perturbed sum ($\sum((x_i + \eta_i) - \bar{x})^2$). $\sum(x_i - \bar{x})^2 = \sum({x_i}^2 - 2\bar{x}x_i + \bar{x}^2)$ and $\sum((x_i + \eta_i) - \bar{x})^2 = \sum {x_i}^2 - 2x_i\bar{x} + 2x_in_i + \bar{x}^2 - 2\bar{x}n_i + {n_i}^2$. These expressions have a lot of shared terms. To determine which is larger, we can cancel many of these shared terms (${x_i}^2$, $2x_i\bar{x}$, and $\bar{x}^2$). Then the perturbed equation remains as $\sum 2x_i\eta_i - 2\bar{x}\eta_i + \eta_i^2$. Note that $\sum 2x_i\eta_i - 2\bar{x}\eta_i = 0$ because $\sum x_i - \bar{x} = 0$. So this will simplify to $\sum \eta_i^2$ which is necessarily greater than 0. So $\sum(x_i - \bar{x})^2 < \sum((x_i + \eta_i) - \bar{x})^2$. So the numerator is unchanged by the jittering, but the denominator is larger. So the expression for $\hat{\beta}_1$ will have an unchanged numerator and a larger denominator, resulting in an underestimation of $\hat{\beta}_1$, or a negative bias.

5. In a residual plot, we plot the residuals e_i vs the fitted values $\hat{y}_i$. Rather than a scatterplot of $(\hat{y}_i, e_i)$, we might have considered (y_i, e_i) or (x_i, e_i). How would they differ and why might we have chosen the form that is conventionally used?

In the case where the data has a positive linear trend, a plot of (y_i, e_i) would be rather uninformative, since the plot will merely show that bigger y_i’s tend to be associated with positive residuals (since larger values of y_i will tend to lie above the line of best fit). This will result in the (y_i, e_i) plot will show an upward line, which is not indicative of the linearity or constant variance. Thus, this graph is not particularly useful to us.

A plot of (x_i, e_i) would allow us to assess the constant variance assumption of the linear model, as we could see whether the residuals were consistent across different values of x_i. The plot would look very similar to the (y_i, e_i) plot, as a strong linear trend with constant variance would show balanced residuals and a horizontal line.

However, we eschew the (x_i, e_i) plot for the plot of the fitted values vs residuals plot $(\hat{y}_i, e_i)$, since this plot works with higher dimensions. In a multivariate scenario, when we have many predictor variables, an (x_i, e_i) plot wouldn’t make sense. This is where the comparative utility of the $(\hat{y}_i, e_i)$ becomes apparent.