# Stat 344 Extra Credit

Trey Tipton

April 28, 2022

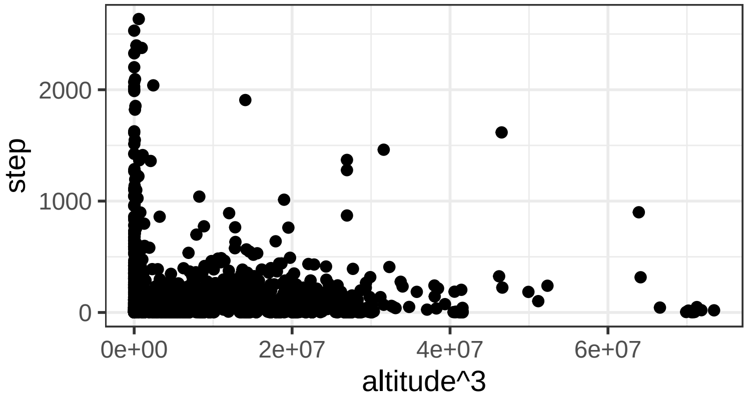
### 3.Yet more oxen

### A. Fit a model

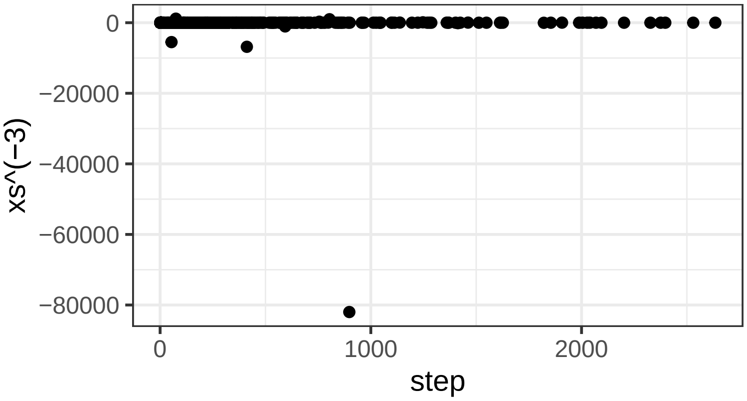
```
ox <- read.csv('https://sldr.netlify.app/data/ox.csv') |>
  mutate(state = factor(state)) |>
  na.omit()

ox <- ox %>%
  mutate(time2 = ifelse(time=="day",1,0))
```

```
gf_point(step ~ altitude^3, data = ox)
```



```
gf_point(xs^(-3) ~ step, data = ox)
```



```
ox_model <- lm(time2 ~ step + step*temp + altitude^3 + xs, data = ox)
msummary(ox_model)
```

```
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  5.569e-01  3.577e-02  15.568  < 2e-16 ***
## step         1.599e-04  4.789e-05   3.340 0.000861 ***
## temp        -1.589e-02  2.114e-03  -7.517 9.94e-14 ***
## altitude    -9.731e-05  1.944e-04  -0.501 0.616662    
## xs          -4.203e-02  2.080e-02  -2.020 0.043536 *  
## step:temp    7.485e-06  6.895e-06   1.086 0.277862    
## 
## Residual standard error: 0.4727 on 1405 degrees of freedom
## Multiple R-squared:  0.05407,    Adjusted R-squared:  0.0507 
## F-statistic: 16.06 on 5 and 1405 DF,  p-value: 2.017e-15
```

For transformations, I did altitude to the third power and xs to the negative third power to balance it out. The scatter plot of the transformed xs variable is a straight line, so the transformation makes sense.

Since the p-value is low, the model is a good fit to the data and model assesment does not need to be done.

$$p_{time} = 3.577e-02 + -4.789e-05x_{step} + 4.789e-05x_{temp} + 1.944e-04x_{altitude} + 2.080e-02x_{xs} + / - 0.0507$$

## Commentary

There is several problems with how this model was created. The question asks to fit a regression model to predict the step length, so step should not be included as a predictor. Likewise, time should not be included as the response variable, especially since it is categorical. The next problems come from the transformations that were made. The justifications are just completely wrong: it is true that you want to make a transformation to see a more linear relationship between response and predictor, but not a horizontal line like in the second plot (not to mention that the x and y are flipped on that plot), and the transformation of xs^(-3) is not even included in the code for lm(). The first transformation makes the data have more of a curve. Plus, if time is your response, it does not make sense to make transformations based on the scatter plots where step is the response variable. There is also no reason to have an interaction between step and temperature, and if that were the case, step should also not be a separate predictor variable if it is included in an interaction. Lastly, the explanation of the p-value is completely wrong. The p-value does not tell you how good of a fit the model is, and it in no way allows model assesment to be skipped because of it.

The model equation has several problems as well. For starters, it uses probability instead of a linear regression model. If it were probability , the equation would most likely need some sort of log function in it for it to work. The equation uses the wrong response variable (time instead of step), and even if time were the response and they were trying to find a probability, it does not signify what the probability is calculating (day or night?). The next problem with the model equation is that the coefficients uses come from the standard error of the model rather than the estimates. Although the transformations in the model were wrong, if this person were to go about transformations, they would need to include them in their model equation as well (example: $p_{time} = x_{altitude}^3 + x_{xs}^{-3}$). This goes for the interaction as well, as there is nothing in the model equation that explains the (incorrect) interaction of step and temperature. The last problem with the model equation is how the error is represented: rather than using the residual standard error, the r-squared value is used incorrectly, and it is treated in the equation as if it is added or subtracted from the equation (as some sort of interval) which is the incorrect way to do so. The residual standard error is represented like so: $y_{step} = 144.2272 + 64.61116x_{time} + ... + \epsilon$, $\epsilon \sim N(0, 0.9229477)$.

### B. Select

Backwards Step-wise Selection: we will do an Anova test on the full model to see which predictor s produce the smallest p-values.

```
car::Anova(ox_model)
```

```
## Anova Table (Type II tests)
##
## Response: time2
##            Sum Sq   Df F value    Pr(>F)
## step        2.317    1 10.3674  0.001312 **
## temp       13.040    1 58.3544 4.028e-14 ***
## altitude    0.056    1  0.2507  0.616662
## xs          0.912    1  4.0818  0.043536 *
## step:temp   0.263    1  1.1784  0.277862
## Residuals 313.960 1405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the full model, step and xs have the smallest p-values, so we should remove them as predictors. This also means that altitude is the best predictor since it has the highest p-value, so we reject the null hypothesis.

```
new_ox <- lm(time2 ~ step*temp + altitude^3, data = ox)

car::Anova(new_ox)
```

```
## Anova Table (Type II tests)
##
## Response: time2
##            Sum Sq   Df F value    Pr(>F)
## step        2.625    1 11.7226 0.0006352 ***
## temp       12.131    1 54.1664 3.121e-13 ***
## altitude    1.506    1  6.7228 0.0096177 **
## step:temp   0.267    1  1.1903 0.2754587
## Residuals 314.872 1406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So now that we removed step and temp, they still have low p-values, but since step:temp interaction produces the largest p-value, we should leave it in. Our final model includes altitude and step:temp as predictors for step.

## Commentary

This looks a lot more like forward selection., but the biggest problem with this selection method is that you are taking out the predictors with the lowest p-values. When doing backwards selection, you have to take out the predictor with the largest large p-value. Also, when doing backwards selection, two predictors should not be removed at the same time; the predictor that produces the largest large p-value should be removed one at a time, then creating the new model and doing the anova test on the new model. The explanation that the altitude is the best predictor simply because it has the largest p-value, which is already wrong, would not necessarily be correct even if it had the lowest p-value. Also, there is no null hypothesis stated, and it would not make sense to have one in this situation or to reject when the p-value is high. Also, the interaction will include p-values separately for step and temp which is why step still had a p-value even after it was removed.

### C. Predict

```r
new_data <- data.frame(altitude = 0, time2 = 1, xs = .5, temp = -10)
conf_int <- predict(ox_model, newdata = new_data,
        interval = 'confidence',
        level = 0.95)
```

```
## Error in model.frame.default(Terms, newdata, na.action = na.action, xlev = object$xlevels): object is
```

```r
conf_int
```

```
## Error in eval(expr, envir, enclos): object 'conf_int' not found
```

A confidence interval for this is impossible which means that there is a probability of zero that there would be an ox at sealevel during night at -10 degrees.

## Commentary

The explanation that the confidence interval is "impossible" does not make sense. In no way does a confidence interval or lack there of produce some sort of probability. In addition, this should be a prediction interval, not a confidence interval, since it is predicting the distance for an individual ox. The reason the code does not run is because the model does not line up with the data frame created. The response variable in the model is wrong, and the interactions and transformations have not been accounted for. Also, the time2 should be equal to zero, not one, since earlier you coded 1 to daytime and 0 to night.