17 Multiple Regression

Katie

April 24, 2017

```r
require(openintro)
```

```
## Loading required package: openintro

## Please visit openintro.org for free statistics materials

##
## Attaching package: 'openintro'

## The following object is masked from 'package:datasets':
##
##     cars
```

```r
data("marioKart")
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```r
require(dplyr)
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

In class today we were thinking about how to predict the price of a mario cart game on Ebay based on several potentially important factors. There is a related dataset called `marioKart` which is included in the package called `openintro`. This dataset contains auction data from Ebay about Nintendo Wii Mario Kart games sold on Ebay. The data was collected in October of 2009.

A first step to working with a new dataset is often to look at the information about the dataset. Type `?marioKart` into the console to look at this information. (Note: we can't include this command in an .rmd file or it will open internet help browsers when we try to knit the file.)

Lets print the top few rows of the dataset here so that we can keep the structure and variable names available for later.

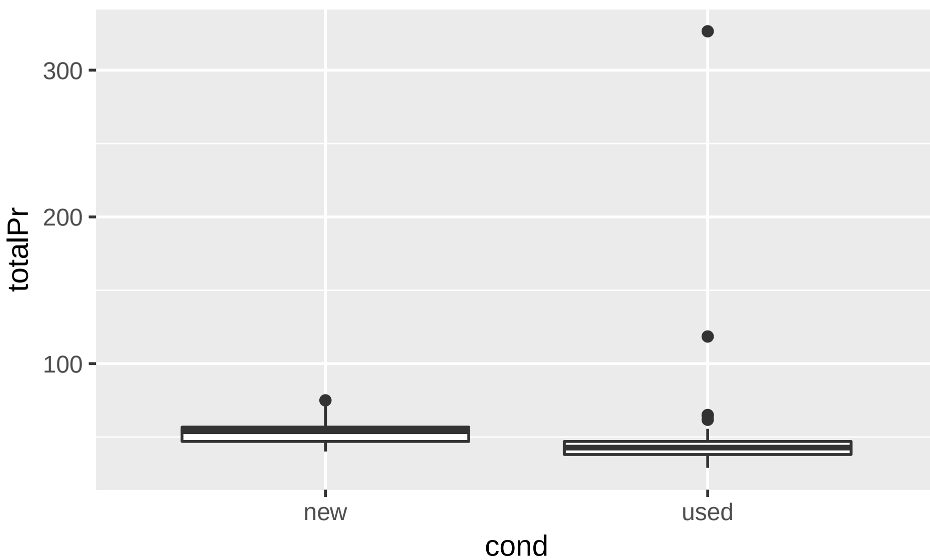
```r
head(marioKart)
```

```
##             ID duration nBids cond startPr shipPr totalPr     shipSp
## 1 150377422259        3    20  new    0.99   4.00   51.55   standard
## 2 260483376854        7    13 used    0.99   3.99   37.04 firstClass
## 3 320432342985        3    16  new    0.99   3.50   45.50 firstClass
## 4 280405224677        3    18  new    0.99   0.00   44.00   standard
## 5 170392227765        1    20  new    0.01   0.00   71.00      media
## 6 360195157625        3    19  new    0.99   4.00   45.00   standard
```

```
##   sellerRate stockPhoto wheels
## 1       1580        yes      1
## 2        365        yes      1
## 3        998         no      1
## 4          7        yes      1
## 5        820        yes      2
## 6     270144        yes      0
##                                                            title
## 1 ~~ Wii MARIO KART &amp; WHEEL ~ NINTENDO Wii ~ BRAND NEW ~~
## 2     Mariokart Wii Nintendo with wheel - Mario Kart Nintendo
## 3                                             Mario Kart Wii (Wii)
## 4        Brand New Mario Kart Wii Comes with Wheel. Free Ship
## 5     BRAND NEW NINTENDO 1 WII MARIO KART WITH 2 WHEELS +GAME
## 6     Mario Kart Wii (GAME ONLY/NO WHEEL) - Nintendo Wii Game
```

Now let's create a boxplot to compare the distribution of the prices between the new and used games.

```
ggplot(data=marioKart, aes(x=cond, y=totalPr))+geom_boxplot()
```



Looking at this boxplot, we see that we have a couple of extreme outliers in the used games that have very high prices. There seems to be something going on that is inflating the price of these games. Let's use the filter function to look only at the two biggest outliers for the used games.

```
marioKart %>%
  filter(cond=="used", totalPr>100)
```

```
##             ID duration nBids cond startPr shipPr totalPr shipSp
## 1 110439174663        7    22 used    1.00  25.51  326.51 parcel
## 2 130335427560        3    27 used    6.95   4.00  118.50 parcel
##   sellerRate stockPhoto wheels
## 1        115         no      2
## 2         41         no      0
##                                                       title
## 1    Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart
## 2 10 Nintendo Wii Games - MarioKart Wii, SpiderMan 3, etc
```

Specifically, lets look at the titles to see if there is anything in the title that might explain their high price. It looks like both of these are bundles containing multiple games. This would be a good reason to remove them from our dataset, but lets also look at the other less extreme outliers to see if any of them should be removed as well.

```
ExpensiveGames <-marioKart %>%
  filter(cond=="used", totalPr>50)
ExpensiveGames$title
```

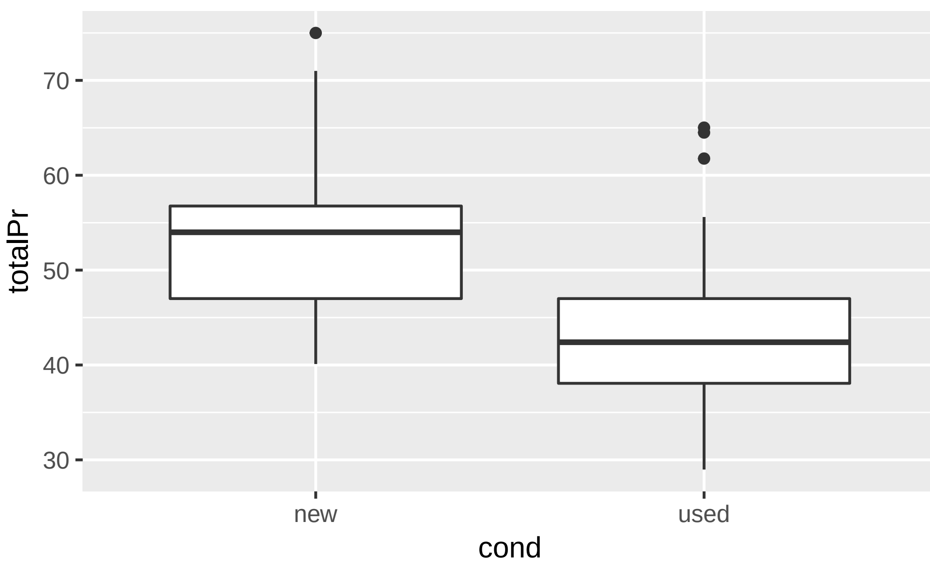
```
##  [1] Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart
##  [2] 10 Nintendo Wii Games - MarioKart Wii, SpiderMan 3, etc
##  [3] Mario Kart Wii (Wii) game and 2 wheels!
##  [4] Wii  MARIO KART GAME  PLUS 3 STEERING WHEELS!   Wii
##  [5] Mario Kart Wii (Wii) + 4 nerf wheels
##  [6] MARIO KART FOR NINTENDO Wii WITH 2 WHEELS + GAME
##  [7] Mario Kart Wii (Wii) w/2 Nintendo Wheels Excellent Cond
##  [8] NINTENDO WII MARIO KART GAME DISK WITH TWO (2) WHEELS
##  [9] Nintendo Wii Mario Kart With 2 Racing Wheels
## [10] Mario Kart Wii with Bonus Wheel!!! (2 Wheels Included)
## [11] Mario Kart Wii (Wii)
## 80 Levels:  Mario Kart Wii with Wii Wheel for Wii (New) ...
```

It looks like all of the used games with prices between \$50 and \$100 are not bundles, so lets just remove the two that cost over \$100.

```
marioKart<- marioKart %>%
  filter(totalPr<100)
```

Now lets look at the boxplot again with these two extreme outliers removed.

```
ggplot(data=marioKart, aes(x=cond, y=totalPr))+geom_boxplot()
```



Is the condition of the game a strong predictor of the total price paid on Ebay?

We want to start by looking at whether the condition of the game (new or used) is strongly correlated to the price. We could answer this question using a t-test or ANOVA but sometimes there are reasons that we would like to instead use linear regression (which is another valid way to analyze the relationship). To do this we can create an *indicator variable* that is a new variable (`cond_new`) in our dataset with a `1` if the game was new and a `0` if the game was used.

```r
marioKart <- marioKart %>%
  mutate(cond_new= (cond=="new")*1)
```

Then use the new indicator variable (`cond_new`) in the `lm()` function as the predictor variable.

```r
fit=lm(totalPr~ cond_new, data=marioKart)
summary(fit)
```

```
##
## Call:
## lm(formula = totalPr ~ cond_new, data = marioKart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8911  -5.8311   0.1289   4.1289  22.1489
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   42.871      0.814  52.668  < 2e-16 ***
## cond_new      10.900      1.258   8.662 1.06e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.371 on 139 degrees of freedom
## Multiple R-squared:  0.3506, Adjusted R-squared:  0.3459
## F-statistic: 75.03 on 1 and 139 DF,  p-value: 1.056e-14
```

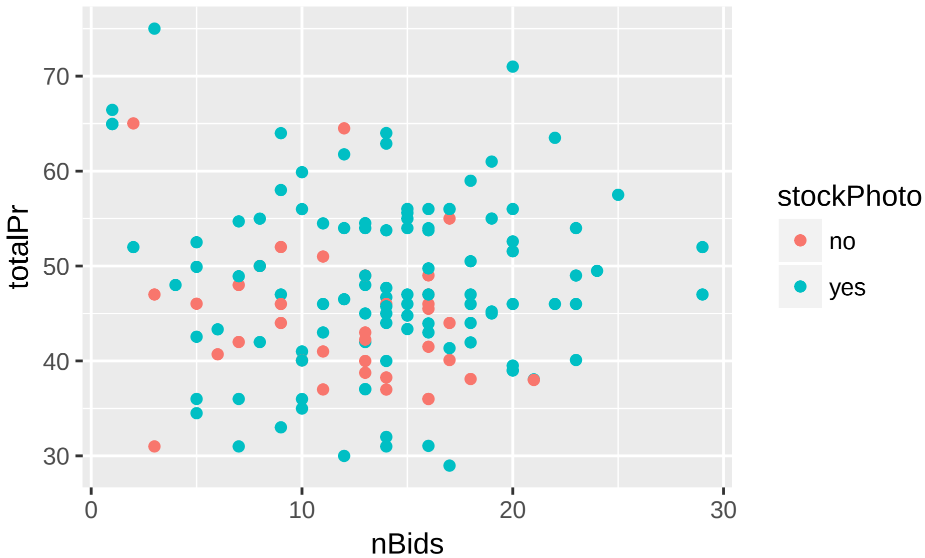
This tells us that the prices of new and used games are significantly different ($t=8.66$, $p<0.001$) and that, according to a linear regression model, on average, a used game costs $42.87, and a new game costs roughly $10.90 more than a used game.

Is the price of Mario Kart dependent on more than just the condition?

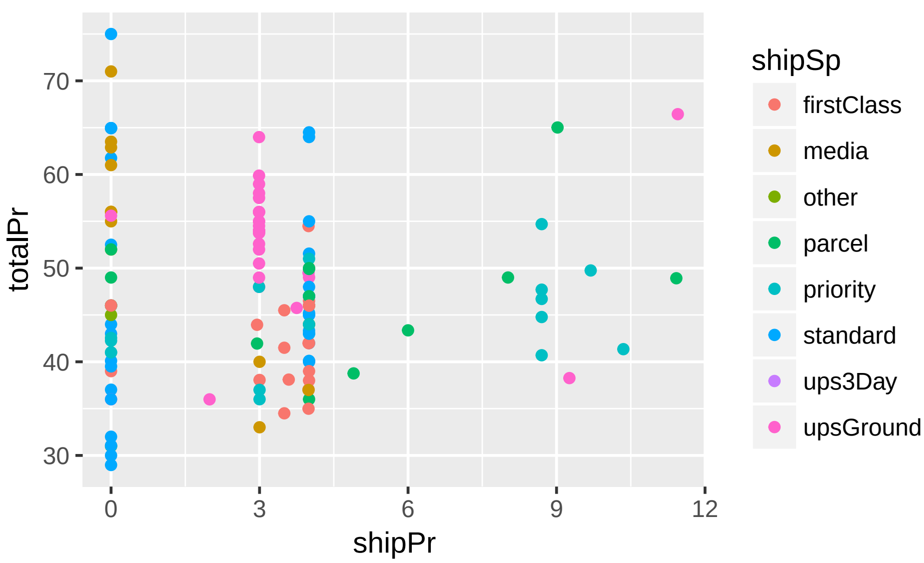
Now, we want to think about what other factors might be important to consider when creating a model for the price of a Mario Kart game. We already know that the condition may be important, but it might also be important to consider the number of bids places (`nBids`), the duration of the auction (`duration`), the shipping speed or method (`shipSp`) and the shipping price (`shipPr`), whether the auction feature photo was a stock photo (`stockPhoto`), and the number of Wii wheels included in the auction (`wheels`).

Let's start by creating some relevant graphics to help us undertand some of these different variables and the possible relationship they might have with the total price someone is willing to pay for the game.

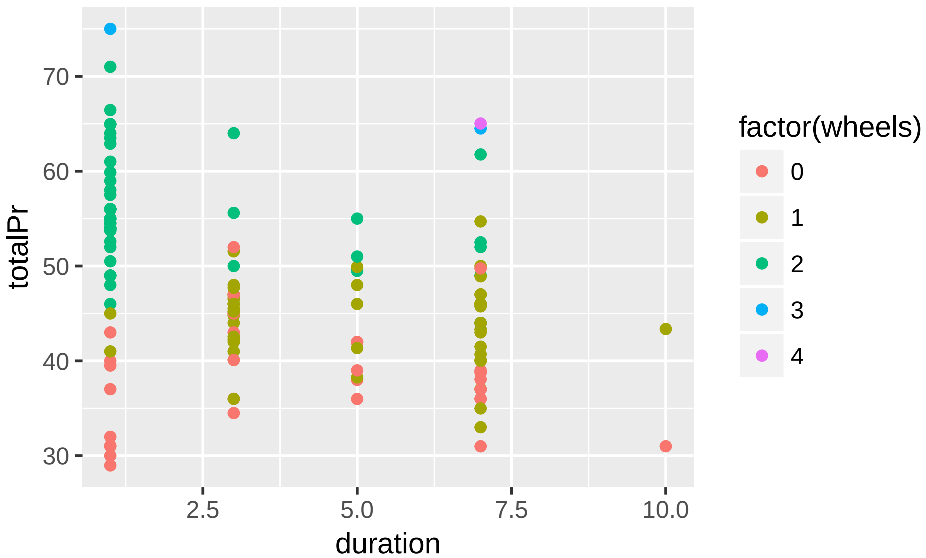
```r
ggplot(data=marioKart, aes(x=nBids, y=totalPr, color=stockPhoto))+geom_point()
```




```
ggplot(data=marioKart, aes(x=shipPr, y=totalPr, color=shipSp))+geom_point()
```




```
ggplot(data=marioKart, aes(x=duration, y=totalPr, color=factor(wheels)))+geom_point()
```



Now we will fit a full model with all reasonable parameters and then use the backward selection method described in your book to pare down the number of variables in the model.

```
fit=lm(totalPr~ cond_new+stockPhoto+nBids+duration+shipSp+wheels+shipPr, data=marioKart)
summary(fit)
```

```
##
## Call:
## lm(formula = totalPr ~ cond_new + stockPhoto + nBids + duration +
##     shipSp + wheels + shipPr, data = marioKart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.5253  -3.2645  -0.3112   3.1378  13.2501
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      37.35395    2.10864  17.715  < 2e-16 ***
## cond_new          5.66776    1.09795   5.162 9.16e-07 ***
## stockPhotoyes     1.18031    1.09146   1.081    0.282
## nBids            -0.11233    0.07652  -1.468    0.145
## duration         -0.19282    0.20565  -0.938    0.350
## shipSpmedia       1.30105    1.84628   0.705    0.482
## shipSpother      -0.97907    3.07232  -0.319    0.750
## shipSpparcel      0.97494    1.73852   0.561    0.576
## shipSppriority   -0.09711    1.48133  -0.066    0.948
## shipSpstandard    0.87516    1.43225   0.611    0.542
## shipSpups3Day    -3.91251    5.05862  -0.773    0.441
## shipSpupsGround  -2.09970    1.62778  -1.290    0.199
## wheels            7.34310    0.60897  12.058  < 2e-16 ***
## shipPr            0.23920    0.18072   1.324    0.188
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.876 on 127 degrees of freedom
## Multiple R-squared:  0.7403, Adjusted R-squared:  0.7138
## F-statistic: 27.86 on 13 and 127 DF,  p-value: < 2.2e-16
```

Now we pick the variable with the highest p-value and elliminate it. There are a number of indicator variables in our model which are associated with the shipping method and they all have high p-values (6 out of the 7 have higher p-values than anything not associated with this variable). So let's start by elliminating the shipping method. If some of the options within shipping method looked like they might be important we could create out own indicator variables and elliminate them one at a time.

```
fit=lm(totalPr~ cond_new+stockPhoto+nBids+duration+wheels+shipPr, data=marioKart)
summary(fit)
```

```
##
## Call:
## lm(formula = totalPr ~ cond_new + stockPhoto + nBids + duration +
##     wheels + shipPr, data = marioKart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6578  -3.1587  -0.7613   2.8454  15.0332
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   37.59848    1.85241  20.297  < 2e-16 ***
## cond_new       5.21031    1.04992   4.963 2.07e-06 ***
## stockPhotoyes  1.18896    1.05575   1.126    0.262
## nBids         -0.11057    0.07327  -1.509    0.134
## duration      -0.09599    0.19696  -0.487    0.627
## wheels         7.13820    0.55902  12.769  < 2e-16 ***
## shipPr         0.13642    0.16377   0.833    0.406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.883 on 134 degrees of freedom
## Multiple R-squared:  0.7252, Adjusted R-squared:  0.7129
## F-statistic: 58.95 on 6 and 134 DF,  p-value: < 2.2e-16
```

Notice that the p-value for the shipPr increased from .188 to .406. Often we see jumps like this when the variable being deleated is strongly correlated with the remaining variable. Here, it is very likely that the shipping method and shipping price will be strongly correlated.

Now, back to elliminating variables. The variable with the largest p-value in the new model is `duration`, so we elliminate it.

```
fit=lm(totalPr~ cond_new+stockPhoto+nBids+wheels+shipPr, data=marioKart)
summary(fit)
```

```
##
## Call:
## lm(formula = totalPr ~ cond_new + stockPhoto + nBids + wheels +
##     shipPr, data = marioKart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9153  -3.1579  -0.9244   2.7964  15.0524
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   37.04234    1.45510  25.457  < 2e-16 ***
```

```
## cond_new       5.36293    0.99930   5.367 3.39e-07 ***
## stockPhotoyes  1.31120    1.02263   1.282    0.202
## nBids         -0.10749    0.07279  -1.477    0.142
## wheels         7.19050    0.54707  13.144  < 2e-16 ***
## shipPr         0.11623    0.15800   0.736    0.463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.869 on 135 degrees of freedom
## Multiple R-squared:  0.7248, Adjusted R-squared:  0.7146
## F-statistic: 71.09 on 5 and 135 DF,  p-value: < 2.2e-16
```

Now elliminate shipPr.

```r
fit=lm(totalPr~ cond_new+stockPhoto+nBids+wheels, data=marioKart)
summary(fit)
```

```
##
## Call:
## lm(formula = totalPr ~ cond_new + stockPhoto + nBids + wheels,
##     data = marioKart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.9272  -3.0540  -0.9173   2.8569  14.7437
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   37.45271    1.34167  27.915  < 2e-16 ***
## cond_new       5.32664    0.99639   5.346  3.7e-07 ***
## stockPhotoyes  1.27197    1.01951   1.248    0.214
## nBids         -0.10998    0.07259  -1.515    0.132
## wheels         7.20230    0.54591  13.193  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.861 on 136 degrees of freedom
## Multiple R-squared:  0.7236, Adjusted R-squared:  0.7155
## F-statistic: 89.03 on 4 and 136 DF,  p-value: < 2.2e-16
```

Now elliminate stockPhoto.

```r
fit=lm(totalPr~ cond_new+nBids+wheels, data=marioKart)
summary(fit)
```

```
##
## Call:
## lm(formula = totalPr ~ cond_new + nBids + wheels, data = marioKart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3851  -3.1625  -0.6905   2.9760  14.8042
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 38.16719    1.21577  31.393  < 2e-16 ***
```

```
## cond_new     5.77457    0.93135   6.200 6.22e-09 ***
## nBids       -0.10094    0.07237  -1.395    0.165
## wheels       7.13639    0.54445  13.107  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.871 on 137 degrees of freedom
## Multiple R-squared:  0.7205, Adjusted R-squared:  0.7144
## F-statistic: 117.7 on 3 and 137 DF,  p-value: < 2.2e-16
```

And now elliminate `nBids`.

```
fit=lm(totalPr~ cond_new+wheels, data=marioKart)
summary(fit)
```

```
##
## Call:
## lm(formula = totalPr ~ cond_new + wheels, data = marioKart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0078  -3.0754  -0.8254   2.9822  14.1646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.7849     0.7066  52.062  < 2e-16 ***
## cond_new      5.5848     0.9245   6.041 1.35e-08 ***
## wheels        7.2328     0.5419  13.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.887 on 138 degrees of freedom
## Multiple R-squared:  0.7165, Adjusted R-squared:  0.7124
## F-statistic: 174.4 on 2 and 138 DF,  p-value: < 2.2e-16
```

This leaves us with the model: $\hat{totalPr} = \$36.78 + \$5.58 \cdot cond\_new + \$7.23 \cdot wheels$. In this model, both the condition of the game and the number of wheels included with the game have statistically significant coefficients, meaning that there is a statistically significant effect of both variables on the price of the game. Overall, there is a strong correlation between the condition and number of wheels and the total price paid for the game, with roughly 72% of the varibility in price being accounted for by a multiple linear model with only these two variables ($R^2_{Adj} = 0.712$).

Does the paper weight or drop height change the distance a paper helicopter falls from the target?

Now we will turn our attention back to the helicopter data that we gathered several weeks ago. At the time, you were simply comparing the distance from the target for drops of your helicopter with a paperclip to helicopter drops without the paperclip. Unbenownst to you, you also were collecting a larger set of data. This data is not perfect in that some of the differences could be differences between the people dropping the helicopter and their accuracy in aiming, but we will essentially ignore this for now and think about the effects of the drop height, paper weight, and paperclip on the distance to the target.

First we need to enter the data:

```
LivAlly=data.frame(
  Weight= rep("L", 20),
```

```r
  DropHeight= rep(44, 20),
  Type= c(rep("NoClip", 10), rep("Clip", 10)),
  Distance= c(c(7.5,2.5,2.5,3.5,5.5,5.25,3.5,9.75,3.75,3.5),
  c(3.5, 3.75, 6.5, 4.0, 7.5, 6.25, 16.5, 6.0, 8.0, 10.25)))
ZStephanieMorgan=data.frame(
    Weight = rep("H",20),
    DropHeight = rep(66,20),
    Type= c(rep("NoClip",10),rep("Clip",10)),
    Distance=c(
      c(6.5, 8.5, 2.5, 14.5, 5, 2, 10, 16, 6.5, 1),
      c(3.5, 1, 17, 12, 6, 5.5, 5.5, 2, 8, 7.5)
    ))
JorjaLiam= data.frame(
    Weight = rep("H",40), ###
    DropHeight = rep(54,40),
    Type= c(rep("NoClip",20),rep("Clip",20)),
    Distance=c(
      c(5.5,    8,  8.5,    12.5,   5.5,    10.5,   9,  6,  2.5,    8.5,    7,  5,  6.5,    12.75,  3.5
      c(5.5,    8,  4,  11, 6.5,    2,  6.5,    5,  5,  11, 7,  5,  2,  1,  5,  8.5,    5.5,    4,  6,
    ))
MelodiFran=data.frame(
  Weight = rep("L", 20),
  DropHeight=rep(67, 20),
  Type = c(rep("NoClip", 10), rep("Clip", 10)),
  Distance = c(c(9.4, 4.9, 6.1, 4.4, 4.5, 8.5, 9, 7.9, 3.8, 4.5), c(3, 2, 5, 0, 2.5, 4.5, 2.5, 2.3, 3.6
AkashKendra = data.frame(
  Weight = rep("H", 20),
  DropHeight = rep(100,20),
  Type = c(rep("Clip", 10), rep("NoClip",10)),
  Distance = c (
    c(22.5,33,20,16,13,9,3,2.5,9.5,3),
    c(14,9,8.5,6,12,9,10.5,10,4,10)
  )
)
SadipGrp=data.frame(
        Weight = rep("L", 20),
        DropHeight = rep(95,20),    # 95 inches
        Type = c(rep("Clip", 10), rep("NoClip", 10)),
        Distance = c (
                c(7.5,3.9,23.6,15.7,11,0.8,15.8,1.2,1.6,5.5),
                c(11.8,14.6,29.9,34.3,25.6,39.4,34.7,29.9,11.8,12.2)
        )
)

#Combine all of the groups datasets to create one dataset with all of the information.
# rbind (row bind) places each dataset below the previous one in the data frame.
PaperClipData=rbind(LivAlly,ZStephanieMorgan,JorjaLiam,MelodiFran,AkashKendra,SadipGrp)
```

Lets look at the first few rows to make sure that we are clear on the variable names and data types.

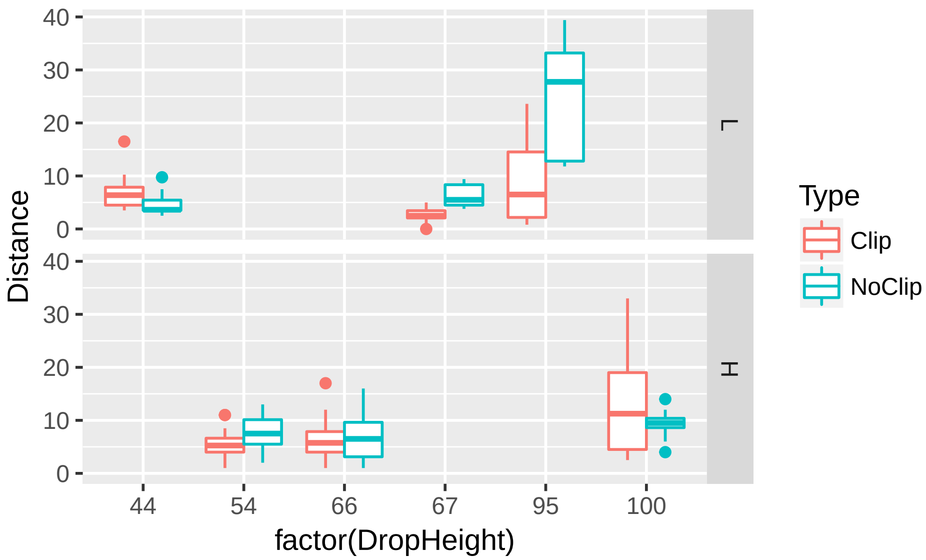
```r
head(PaperClipData)
```

```
##   Weight DropHeight   Type Distance
## 1      L         44 NoClip     7.50
```

```
## 2       L          44 NoClip     2.50
## 3       L          44 NoClip     2.50
## 4       L          44 NoClip     3.50
## 5       L          44 NoClip     5.50
## 6       L          44 NoClip     5.25
```

Now lets create some helpful graphs. It would be nice to include several explanitory variables in our graph. This is one way we could include all of them. Note: it is not as clear because the heights for the two groups do not exactly line up. In retrospect, it would have been nice to have a variable to store whether the dropper was standing, on a chair or 6 or 9 steps up the stairs.

```
ggplot(data=PaperClipData, aes(x=factor(DropHeight), y=Distance, colour=Type))+geom_boxplot()+facet_gri
```



Creating the multiple linear model using forward selection

We will start by fitting all possible linear models with one explanitory variable. We will pick the one with the highest adjusted R^2 value and then look at adding a second variable.

The model with `DropHeight` as the explanitory variable.

```
fit=lm(Distance~ DropHeight, data=PaperClipData)
summary(fit)
```

```
##
## Call:
## lm(formula = Distance ~ DropHeight, data = PaperClipData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6763  -3.7185  -0.7657   2.1986  26.9237
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.35227    1.98023  -1.188    0.237
## DropHeight   0.15609    0.02775   5.624 9.97e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 6.477 on 138 degrees of freedom
## Multiple R-squared:  0.1865, Adjusted R-squared:  0.1806
## F-statistic: 31.63 on 1 and 138 DF,  p-value: 9.971e-08
```

The model with `Weight` as the explanitory variable.

```
fit=lm(Distance~ Weight, data=PaperClipData)
summary(fit)
```

```
##
## Call:
## lm(formula = Distance ~ Weight, data = PaperClipData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.990 -4.872 -1.872  2.128 30.410
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.9900     0.9243   9.726   <2e-16 ***
## WeightH      -1.1181     1.2228  -0.914    0.362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.16 on 138 degrees of freedom
## Multiple R-squared:  0.006023,   Adjusted R-squared:  -0.00118
## F-statistic: 0.8361 on 1 and 138 DF,  p-value: 0.3621
```

The model with `Type` as the explanitory variable.

```
fit=lm(Distance~ Type, data=PaperClipData)
summary(fit)
```

```
##
## Call:
## lm(formula = Distance ~ Type, data = PaperClipData)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.624 -4.578 -1.578  1.536 29.776
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.0779     0.8446   8.380 5.41e-14 ***
## TypeNoClip    2.5464     1.1944   2.132   0.0348 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.066 on 138 degrees of freedom
## Multiple R-squared:  0.03189,    Adjusted R-squared:  0.02487
## F-statistic: 4.545 on 1 and 138 DF,  p-value: 0.03478
```

The model with the highest R^2_{Adj} value models the Distance as a function of `DropHeight`. So we will start there and look at all the possible models with a second variable added.

```r
fit=lm(Distance~ DropHeight+Type, data=PaperClipData)
summary(fit)
```

```
##
## Call:
## lm(formula = Distance ~ DropHeight + Type, data = PaperClipData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5300  -3.8949  -0.8644   2.3100  25.6505
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.6255     2.0212  -1.794   0.0751 .
## DropHeight    0.1561     0.0273   5.717  6.5e-08 ***
## TypeNoClip    2.5464     1.0771   2.364   0.0195 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.372 on 137 degrees of freedom
## Multiple R-squared:  0.2184, Adjusted R-squared:  0.207
## F-statistic: 19.14 on 2 and 137 DF,  p-value: 4.685e-08
```

```r
fit=lm(Distance~ DropHeight+Weight, data=PaperClipData)
summary(fit)
```

```
##
## Call:
## lm(formula = Distance ~ DropHeight + Weight, data = PaperClipData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2974  -3.7851  -0.8076   2.3898  26.3026
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.72034    2.08130  -0.827    0.410
## DropHeight   0.15598    0.02776   5.620 1.03e-07 ***
## WeightH     -1.09213    1.10634  -0.987    0.325
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.478 on 137 degrees of freedom
## Multiple R-squared:  0.1922, Adjusted R-squared:  0.1804
## F-statistic:  16.3 on 2 and 137 DF,  p-value: 4.461e-07
```

The model with `Type` added has a larger R^2_{Adj} value than the model without it and larger than with `Weight` added so we add igb `Type` to our model. Then we look at what happens if we add `Weight` as a third variable in the model.

```r
fit=lm(Distance~ DropHeight+Type+Weight, data=PaperClipData)
summary(fit)
```

```
##
## Call:
## lm(formula = Distance ~ DropHeight + Type + Weight, data = PaperClipData)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0242  -3.9514  -0.8834   2.2174  25.0294
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.9936     2.1169  -1.414   0.1596
## DropHeight    0.1560     0.0273   5.713 6.71e-08 ***
## TypeNoClip    2.5464     1.0771   2.364   0.0195 *
## WeightH      -1.0921     1.0883  -1.004   0.3174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.372 on 136 degrees of freedom
## Multiple R-squared:  0.2241, Adjusted R-squared:  0.207
## F-statistic: 13.09 on 3 and 136 DF,  p-value: 1.456e-07
```

This does not improve the R^2_{Adj} value so we won't include weight in our model.

Our best multiple linear model for the distance to the target accounts for the drop height (inches) and the type (paperclip or no paperclip) and both of these variables have coefficients that are significantly different from 0 (p<0.001 for the drop height coeeficient and p=0.0195 for the Type coefficient). The model is

$$\hat{Distance} = -3.63 + 0.156 DropHeight + 2.546 TypeNoClip$$

Which means that on average, each increase in the drop height by 1 inche increases the distance to the target by 0.156 inches, and the NoClip helicopters fall, on average, 2.5 inches further from the target than the Clip helicopters. This model is statistically significant but these two variables only explain roughly 22% of the variance in the distance and because of this, the model does not have great predictive power ($R^2_{Adj} = 0.207$). Most of this comes from the fact that there is a great deal of variability inherrant in our data and this will never be explained fully by almost any model. This is the same problem that often arrises in ecological data and is the reason that R^2 values close to .3 often show a fairly strong relationship where R^2 values close to .3 in almost any other field would be showing a weak association between the variables.