SLR

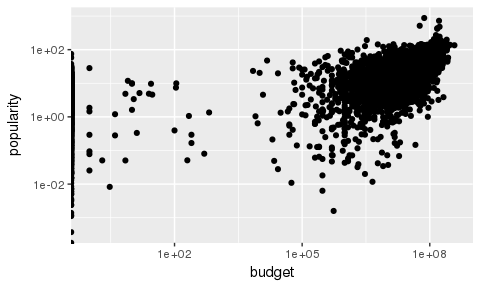
Maryann Zhao and Leyla Akay

## Introduction

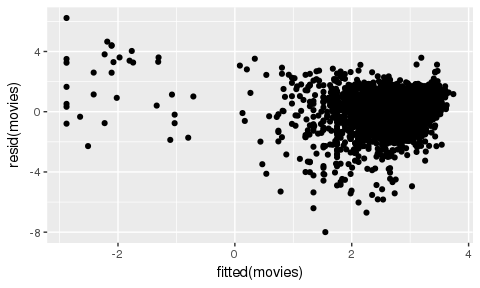
We are studying a dataset of movies, that contains information about their languages, actors, genres, etc. For this project, our variables of interest are budget (in US Dollars) and popularity (based on number of votes per day, and number of views and favorites). We are interested in determining whether there is a linear relationship between the two, and if so, the nature of the relationship. Our null hypothesis is that there is no relationship between budget and popularity; our alternative hypothesis is that there is a positive relationship between budget and popularity. That is, .

To test this, first we must check that the data appears to fit with the technical assumptions necessary to perform a linear regression. The first is linearity. An initial plot of budget values vs. popularity indicates a linear shape. We transformed both explanatory and response variables with a log function, and noticed that the plot became more linear, so we decided to keep the transformation for the remaining tests. The second is independence--we are assuming mathematical independence. That is, spending more money doesn't necessarily translate to increased popularity, or vice versa. The final assumptions are normality and constant errors. To check this, we plotted the residuals and looked at the symmetry. The residuals appear to be symmetrically distributed around 0, with some increased scattering at lower values of budget. The residuals appear more concentrated around 0, and are all in the range of +/- 4.

#plot of log popularity vs. log budget  
ggplot(tmdb2, aes(x=budget, y=popularity)) +  
 geom\_point() +  
 scale\_x\_log10() +  
 scale\_y\_log10()



#residual plot  
bud2 <- tmdb2 %>%   
 select(popularity, budget) %>%  
 filter(budget > 0, popularity > 0)  
pop.log <- log(bud2$popularity)  
bud.log <- log(bud2$budget)  
movies <- lm(pop.log ~ bud.log, data = tmdb2)  
ggplot (movies, aes(x=fitted(movies), y=resid(movies))) + geom\_point()

 ##Testing To determine whether the alternative hypothesis holds, we found a 95% confidence interval for . We found the CI was between 0.317 and 0.353. Because both variables had been logged-transformed, this means that a doubling of budget would be consistent with a multiplicative change in median of popularity between: , which is consistent with 1.25 and 1.28.

We were interested in movies with 1M$ budgets, since these are relatively low compared to many block-buster films, but many appear to achieve comparable levels of popularity. In this situation, we thought of budget as the explanatory variable, since it's determined before popularity, so although budget doesn't necessarily explain popularity, it wouldn't be possible for popularity to inform budget.

#CI and PI for 1 mill budget  
broom::tidy(movies, conf.int = TRUE, conf.level = 0.95)

## term estimate std.error statistic p.value conf.low conf.high  
## 1 (Intercept) -2.881 0.15287 -18.8 8.99e-76 -3.180 -2.581  
## 2 bud.log 0.335 0.00916 36.6 3.51e-251 0.317 0.353

mill <- data.frame(bud.log=c(13.8))  
crit\_val <- qt(.95, glance(movies)$df.resid)  
movie.pred <- broom::augment(movies, newdata=mill, type.predict = "response")  
.se.pred <- sqrt(glance(movies)$sigma^2 + movie.pred$.se.fit)  
  
movie.pred <- movie.pred %>%  
mutate(lower\_CI = .fitted - crit\_val \* .se.fit,  
upper\_CI = .fitted + crit\_val \* .se.fit)  
  
movie.pred <- movie.pred %>%  
 mutate(lower\_PI = .fitted - crit\_val \* .se.pred,  
 upper\_PI = .fitted + crit\_val \* .se.pred,   
 lower\_CI = .fitted - crit\_val \* .se.fit,   
 upper\_CI = .fitted + crit\_val \* .se.fit)  
  
movie.pred

## bud.log .fitted .se.fit lower\_CI upper\_CI lower\_PI upper\_PI  
## 1 13.8 1.74 0.0312 1.69 1.79 -0.106 3.59

## Assessing the fit

The r-squared value is 0.262. The residuals (plotted above) appear to be evenly distributed. Therefore, we believe our model accurately describes the data.

#R squared value  
summary(movies)$r.squared

## [1] 0.262

## Conclusion

In general, the positive relationship was expected. The decreased variability at very high budgets was interesting, but upon further reflection it makes sense. Movies that spend tens of millions of dollars probably attract high-profile actors and have lots of advertising campaigns, which probably increase the popularity.

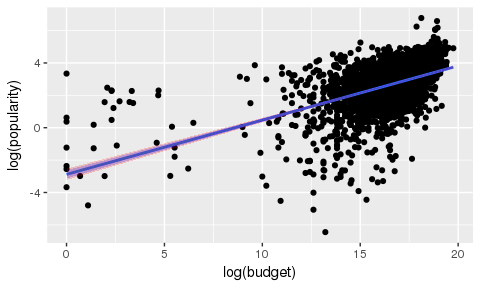
## Pairs: Simultaneous Inference

To find the mean and prediction intervals for all n values, we used the Bonferroni, Working-Hotelling, and Scheffé methods.

#critical values for 3 methods  
num\_int <- 3  
n <- 3766  
crit\_Bonf <- qt((1-.05)/num\_int, glance(movies)$df.resid)  
crit\_WH <- sqrt(2\*qf(.95, num\_int, glance(movies)$df.resid))  
crit\_Sch <- sqrt(num\_int\*qf(.95, n-2, glance(movies)$df.resid))  
  
#mean intervals for all n points  
movie\_gl <- broom::glance(movies)  
movie\_sig <- dplyr::pull(movie\_gl, sigma)  
  
movie\_CI <-broom::augment(movies) %>%  
 mutate(.se.pred = sqrt(movie\_sig^2 + .se.fit^2)) %>%  
 mutate(lower\_CI = .fitted - crit\_val \* .se.fit,  
 upper\_CI = .fitted + crit\_val \* .se.fit,   
 lower\_CI\_B = .fitted - crit\_Bonf\*.se.fit,  
 upper\_CI\_B = .fitted + crit\_Bonf\*.se.fit,   
 lower\_CI\_WH = .fitted - crit\_WH\*.se.fit,  
 upper\_CI\_WH = .fitted + crit\_WH\*.se.fit,  
 lower\_CI\_S = .fitted - crit\_Sch\*.se.fit,  
 upper\_CI\_S = .fitted + crit\_Sch\*.se.fit)  
  
movie\_CI %>% head()

## pop.log bud.log .fitted .se.fit .resid .hat .sigma .cooksd  
## 1 5.01 19.3 3.58 0.0307 1.433 0.000763 1.11 6.36e-04  
## 2 4.94 19.5 3.66 0.0324 1.275 0.000853 1.11 5.64e-04  
## 3 4.68 19.3 3.59 0.0309 1.084 0.000775 1.11 3.70e-04  
## 4 4.72 19.3 3.60 0.0311 1.122 0.000783 1.11 4.01e-04  
## 5 3.78 19.4 3.61 0.0313 0.171 0.000797 1.11 9.43e-06  
## 6 4.75 19.4 3.61 0.0313 1.142 0.000794 1.11 4.21e-04  
## .std.resid .se.pred lower\_CI upper\_CI lower\_CI\_B upper\_CI\_B lower\_CI\_WH  
## 1 1.291 1.11 3.53 3.63 3.60 3.57 3.51  
## 2 1.149 1.11 3.61 3.71 3.68 3.64 3.59  
## 3 0.977 1.11 3.54 3.64 3.61 3.58 3.52  
## 4 1.012 1.11 3.55 3.65 3.61 3.58 3.53  
## 5 0.154 1.11 3.56 3.66 3.63 3.60 3.54  
## 6 1.029 1.11 3.56 3.66 3.62 3.59 3.54  
## upper\_CI\_WH lower\_CI\_S upper\_CI\_S  
## 1 3.65 3.53 3.64  
## 2 3.73 3.60 3.72  
## 3 3.66 3.54 3.65  
## 4 3.67 3.54 3.65  
## 5 3.68 3.56 3.67  
## 6 3.68 3.55 3.67

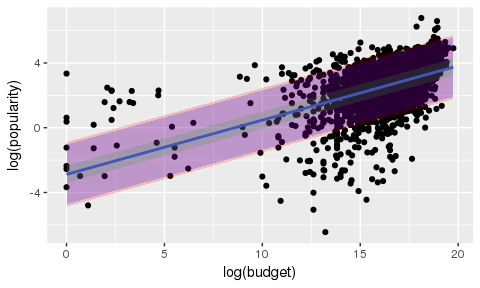
#blue - no adjustment, red - Working-Hotelling, green - Bonferroni  
ggplot(movie\_CI, aes(x = bud.log, y =pop.log)) + geom\_point() +  
 stat\_smooth(method = "lm", se = FALSE) +  
 geom\_ribbon(data=movie\_CI, aes(ymin = lower\_CI\_B, ymax = upper\_CI\_B), alpha = .2, fill = "green") +  
 geom\_ribbon(data = movie\_CI, aes(ymin = lower\_CI\_WH, ymax = upper\_CI\_WH), alpha = .2, fill = "red") +  
 geom\_ribbon(aes(ymin = lower\_CI, ymax = upper\_CI), alpha = .2, fill = "blue") +  
 xlab("log(budget)") + ylab("log(popularity)")



#prediction interval for all n points  
movie\_PI <-broom::augment(movies) %>%  
 mutate(.se.pred = sqrt(movie\_sig^2 + .se.fit^2)) %>%  
 mutate(lower\_PI = .fitted - crit\_val \* .se.pred,  
 upper\_PI = .fitted + crit\_val \* .se.pred,   
 lower\_PI\_B = .fitted - crit\_Bonf\*.se.pred,  
 upper\_PI\_B = .fitted + crit\_Bonf\*.se.pred,   
 lower\_PI\_WH = .fitted - crit\_WH\*.se.pred,  
 upper\_PI\_WH = .fitted + crit\_WH\*.se.pred,  
 lower\_PI\_S = .fitted - crit\_Sch\*.se.pred,  
 upper\_PI\_S = .fitted + crit\_Sch\*.se.pred)  
  
movie\_PI %>% head()

## pop.log bud.log .fitted .se.fit .resid .hat .sigma .cooksd  
## 1 5.01 19.3 3.58 0.0307 1.433 0.000763 1.11 6.36e-04  
## 2 4.94 19.5 3.66 0.0324 1.275 0.000853 1.11 5.64e-04  
## 3 4.68 19.3 3.59 0.0309 1.084 0.000775 1.11 3.70e-04  
## 4 4.72 19.3 3.60 0.0311 1.122 0.000783 1.11 4.01e-04  
## 5 3.78 19.4 3.61 0.0313 0.171 0.000797 1.11 9.43e-06  
## 6 4.75 19.4 3.61 0.0313 1.142 0.000794 1.11 4.21e-04  
## .std.resid .se.pred lower\_PI upper\_PI lower\_PI\_B upper\_PI\_B lower\_PI\_WH  
## 1 1.291 1.11 1.75 5.41 4.11 3.05 1.05  
## 2 1.149 1.11 1.83 5.49 4.19 3.13 1.12  
## 3 0.977 1.11 1.77 5.42 4.12 3.06 1.06  
## 4 1.012 1.11 1.77 5.43 4.13 3.07 1.06  
## 5 0.154 1.11 1.79 5.44 4.14 3.08 1.08  
## 6 1.029 1.11 1.78 5.44 4.14 3.08 1.07  
## upper\_PI\_WH lower\_PI\_S upper\_PI\_S  
## 1 6.12 1.61 5.56  
## 2 6.20 1.68 5.64  
## 3 6.13 1.62 5.57  
## 4 6.13 1.62 5.57  
## 5 6.15 1.64 5.59  
## 6 6.14 1.63 5.58

#blue - no adjustment, red - Scheffer, green - Bonferroni  
ggplot(movie\_PI, aes(x = bud.log, y =pop.log)) + geom\_point() +  
 stat\_smooth(method = "lm", se = FALSE) +  
 geom\_ribbon(data=movie\_PI, aes(ymin = lower\_PI\_B, ymax = upper\_PI\_B), alpha = .2, fill = "green") +  
 geom\_ribbon(data = movie\_PI, aes(ymin = lower\_PI\_S, ymax = upper\_PI\_S), alpha = .2, fill = "red") +  
 geom\_ribbon(aes(ymin = lower\_PI, ymax = upper\_PI), alpha = .2, fill = "blue") +  
 xlab("log(budget)") + ylab("log(popularity)")



It's important to control for multiple comparisons, because the more variables you compare, the more likely you are to get a "significant" difference, even though there might not be a real effect. For this reason, it's important to have stricter measures of confidence when testing multiple variables. For mean intervals, it's most useful to use the Working-Hotelling procedure, because it uses the entire range of x's, whereas the Bonferroni only uses the intervals of interest. However, for prediction intervals, Bonferroni is better than Scheffe because it takes into account the multiple comparisons error.