Multiple Regression Notes

Lt Col Ken Horton Professor Bradley Warner

31 July, 2020

Objectives

1) Create and interpret a model with multiple predictors.
2) Generate and interpret confidence intervals for estimates.
3) Explain adjusted R^2 and and multi-collinearity.
4) Interpret regression coefficients for a linear model with multiple predictors.

Introduction to multiple regression

The principles of simple linear regression lay the foundation for more sophisticated regression methods used in a wide range of challenging settings. In our last two lessons, we explore multiple regression, which introduces the possibility of more than one predictor.

Multiple regression

Multiple regression extends simple two-variable regression to the case that still has one response but many predictors (denoted x_1, x_2, x_3, ...). The method is motivated by scenarios where many variables may be simultaneously connected to an output.

We will consider Ebay auctions of a video game called **Mario Kart** for the Nintendo Wii. The outcome variable of interest is the total price of an auction, which is the highest bid plus the shipping cost. We will try to determine how total price is related to each characteristic in an auction while simultaneously controlling for other variables. For instance, all other characteristics held constant, are longer auctions associated with higher or lower prices? And, on average, how much more do buyers tend to pay for additional Wii wheels (plastic steering wheels that attach to the Wii controller) in auctions? Multiple regression will help us answer these and other questions.

The data set `mariokart` includes results from 141 auctions.[1] Ten observations from this data set are shown in the `R` code below. Multiple regression also allows for categorical variables with many levels, though we do not have any such variables in this analysis, and leave it for a regression course.

```
head(mariokart)
```

```
## # A tibble: 6 x 12
##        id duration n_bids cond  start_pr ship_pr total_pr ship_sp seller_rate
##     <dbl>    <int>  <int> <fct>    <dbl>   <dbl>    <dbl> <fct>         <int>
```

[1] Diez DM, Barr CD, and Çetinkaya-Rundel M. 2012. `openintro`: OpenIntro data sets and supplemental functions. http://cran.r-project.org/web/packages/openintro

```
## 1 1.50e11        3     20 new       0.99    4        51.6 standa~        1580
## 2 2.60e11        7     13 used      0.99    3.99     37.0 firstC~         365
## 3 3.20e11        3     16 new       0.99    3.5      45.5 firstC~         998
## 4 2.80e11        3     18 new       0.99    0        44   standa~           7
## 5 1.70e11        1     20 new       0.01    0        71   media            820
## 6 3.60e11        3     19 new       0.99    4        45   standa~      270144
## # ... with 3 more variables: stock_photo <fct>, wheels <int>, title <fct>
```

We are only interested in `total_pr`, `cond`, `stock_photo`, `duration`, and `wheels`. These variables are described in the list below.

1. `total_pr`: final auction price plus shipping costs, in US dollars
2. `cond`: a two-level categorical factor variable
3. `stock_photo`: a two-level categorical factor variable
4. `duration`: the length of the auction, in days, taking values from 1 to 10
5. `wheels`: the number of Wii wheels included with the auction (a **Wii wheel** is a plastic racing wheel that holds the Wii controller and is an optional but helpful accessory for playing Mario Kart)

A single-variable model for the Mario Kart data

Let's fit a linear regression model with the game's condition as a predictor of auction price.

```
mario_mod <- lm(total_pr~cond,data=mariokart)
```

```
summary(mario_mod)
```

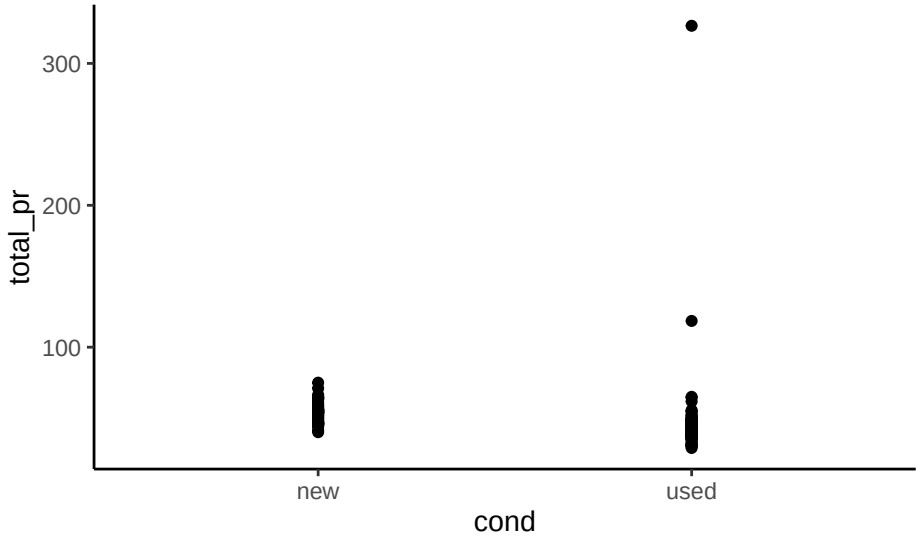
```
##
## Call:
## lm(formula = total_pr ~ cond, data = mariokart)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.168  -7.771  -3.148   1.857 279.362
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   53.771      3.329  16.153   <2e-16 ***
## condused      -6.623      4.343  -1.525     0.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.57 on 141 degrees of freedom
## Multiple R-squared:  0.01622,    Adjusted R-squared:  0.009244
## F-statistic: 2.325 on 1 and 141 DF,  p-value: 0.1296
```

The model may be written as

$$\hat{total price} = 53.771 - 6.623 \times condused$$

A scatterplot for price versus game condition is shown below.

```
mariokart %>%
  gf_point(total_pr~cond) %>%
  gf_theme(theme_classic())
```




```
mariokart %>%
  group_by(cond) %>%
  summarize(xbar=mean(total_pr), stand_dev=sd(total_pr))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   cond   xbar stand_dev
##   <fct> <dbl>     <dbl>
## 1 new    53.8      7.44
## 2 used   47.1     32.7
```

There two outliers in the plot. Let's gather more information about them.

```
mariokart %>%
  filter(total_pr > 100)
```

```
## # A tibble: 2 x 12
##         id duration n_bids cond  start_pr ship_pr total_pr ship_sp seller_rate
##      <dbl>    <int>  <int> <fct>    <dbl>   <dbl>    <dbl> <fct>         <int>
## 1 1.10e11        7     22 used      1       25.5     327. parcel          115
## 2 1.30e11        3     27 used      6.95     4       118. parcel           41
## # ... with 3 more variables: stock_photo <fct>, wheels <int>, title <fct>
```

If you look at the variable `title` there were additional items in the sale. Let's remove them.

```r
mariokart_new <- mariokart %>%
  filter(total_pr <= 100) %>%
  select(total_pr,cond,stock_photo,duration,wheels)
```

```r
mario_mod2 <- lm(total_pr~cond,data=mariokart_new)
```

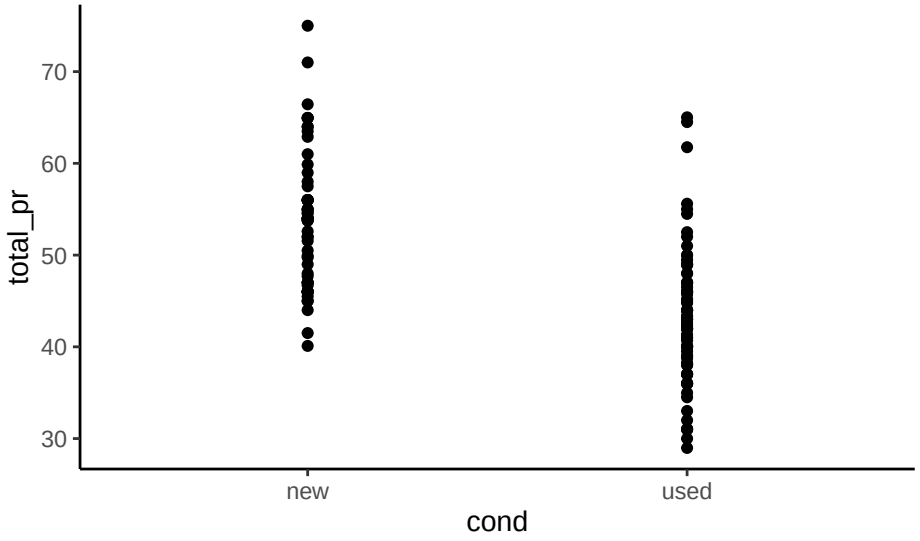
```r
summary(mario_mod2)
```

```
##
## Call:
## lm(formula = total_pr ~ cond, data = mariokart_new)
##
## 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)  53.7707     0.9596  56.034  < 2e-16 ***
## condused    -10.8996     1.2583  -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
```

The model may be written as

$$\hat{total price} = 53.771 - 10.90 \times condused$$

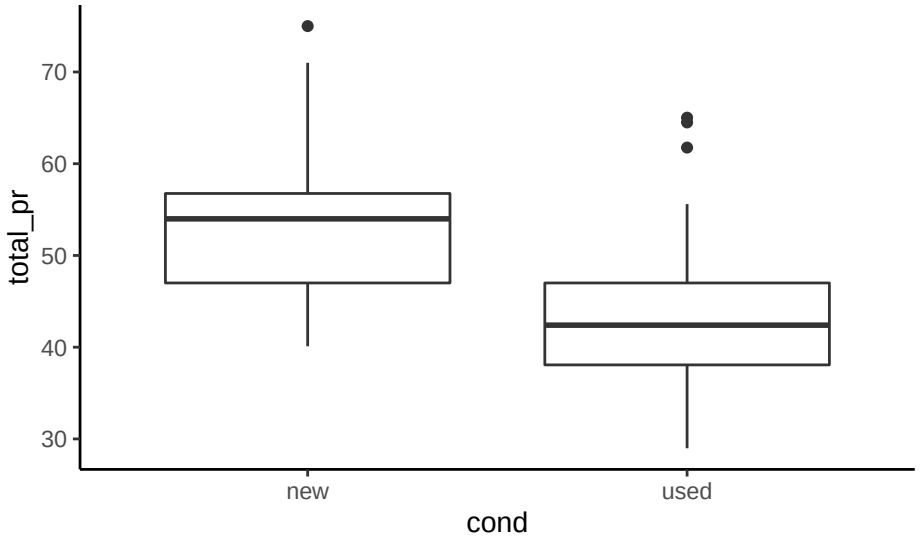
```r
mariokart_new %>%
  gf_point(total_pr~cond) %>%
  gf_theme(theme_classic())
```



```
summary(mariokart_new)
```

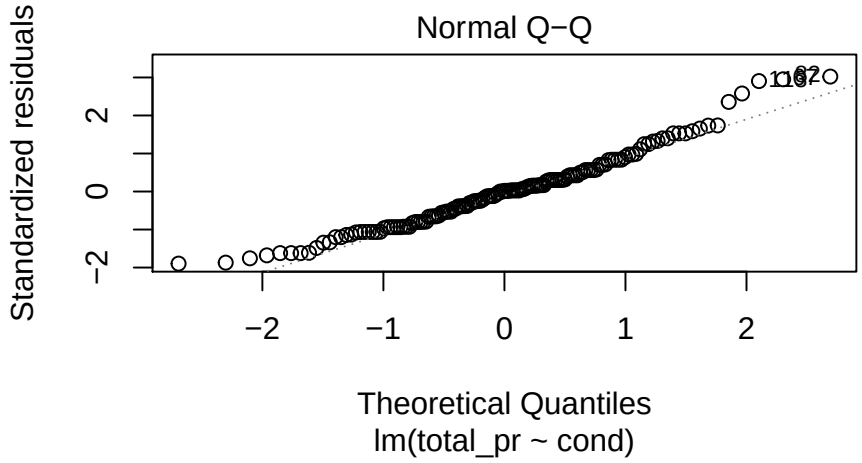
```
##     total_pr       cond    stock_photo    duration          wheels
##  Min.   :28.98   new :59   no : 36     Min.   : 1.000   Min.   :0.000
##  1st Qu.:41.00   used:82   yes:105     1st Qu.: 1.000   1st Qu.:0.000
##  Median :46.03                         Median : 3.000   Median :1.000
##  Mean   :47.43                         Mean   : 3.752   Mean   :1.149
##  3rd Qu.:53.99                         3rd Qu.: 7.000   3rd Qu.:2.000
##  Max.   :75.00                         Max.   :10.000   Max.   :4.000
```

```
mariokart_new %>%
  gf_boxplot(total_pr~cond) %>%
  gf_theme(theme_classic())
```



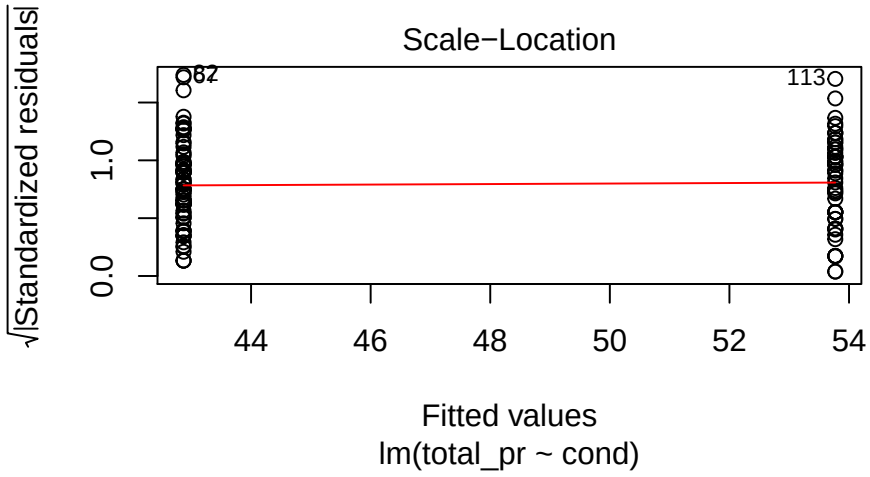
Exercise:
Does the linear model seem reasonable? Which assumptions should you check?

```
plot(mario_mod2,2)
```



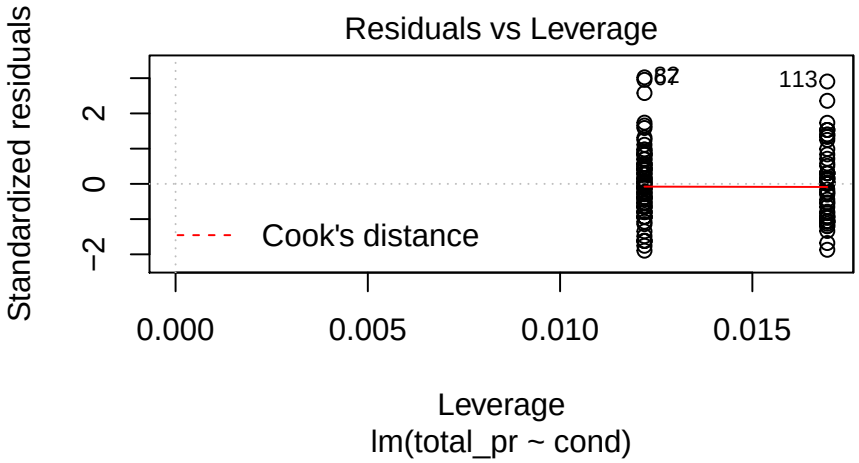
This somewhat suspect but we have more than 100 data points so the short tails of the distribution are not a concern.

```
plot(mario_mod2,3)
```



Equal variance seems reasonable.

```
plot(mario_mod2,5)
```



No high leverage points.

No need to check linearity, we only have two different values for the explanatory variable.

> *Example*: Interpretation
> Interpret the coefficient for the game's condition in the model. Is this coefficient significantly different from 0?

Note that since `cond` is a two-level categorical variable and the reference level is `new`. So - 10.90 means that the model predicts an extra $10.90 for those games that are new versus those that are used. Examining the regression output, we can see that the p-value for `cond` is very close to zero, indicating there is strong evidence that the coefficient is different from zero when using this simple one-variable model.

Including and assessing many variables in a model

Sometimes there are underlying structures or relationships between predictor variables. For instance, new games sold on Ebay tend to come with more Wii wheels, which may have led to higher prices for those auctions. We would like to fit a model that includes all potentially important variables simultaneously. This would help us evaluate the relationship between a predictor variable and the outcome while controlling for the potential influence of other variables. This is the strategy used in **multiple regression**. While we remain cautious about making any causal interpretations using multiple regression, such models are a common first step in providing evidence of a causal connection.

We want to construct a model that accounts for not only the game condition, but simultaneously accounts for three other variables: `stock_photo`, `duration`, and `wheels`.

$$\widehat{totalprice} = \beta_0 + \beta_1 \times cond + \beta_2 \times \text{stockphoto} + \beta_3 \times duration + \beta_4 \times wheels$$

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

In this equation, y represents the total price, x_1 indicates whether the game is new, x_2 indicates whether a stock photo was used, x_3 is the duration of the auction, and x_4 is the number of Wii wheels included with the game. Just as with the single predictor case, a multiple regression model may be missing important components or it might not precisely represent the relationship between the outcome and the available explanatory variables. While no model is perfect, we wish to explore the possibility that this one may fit the data reasonably well.

We estimate the parameters β_0, β_1, ..., β_4 in the same way as we did in the case of a single predictor. We select b_0, b_1, ..., b_4 that minimize the sum of the squared residuals:

$$\text{SSE} = e_1^2 + e_2^2 + \cdots + e_{141}^2 = \sum_{i=1}^{141} e_i^2 = \sum_{i=1}^{141} (y_i - \hat{y}_i)^2$$

Here there are 141 residuals, one for each observation. We use a computer to minimize the sum and compute point estimates.

```
mario_mod_multi <- lm(total_pr~., data=mariokart_new)
```

The formula `total_pr~.` uses a *dot*. This means we want to use all the predictors. We could have also used the following code:

```
mario_mod_multi <- lm(total_pr~cond+stock_photo+duration+wheels, data=mariokart_new)
```

The `+` symbol does mean to literally add the predictors together. It is not a mathematical operation but a formula operation that means to include the predictor.

```
summary(mario_mod_multi)
```

```
##
## Call:
## lm(formula = total_pr ~ ., data = mariokart_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3788  -2.9854  -0.9654   2.6915  14.0346
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    41.34153    1.71167  24.153  < 2e-16 ***
## condused       -5.13056    1.05112  -4.881 2.91e-06 ***
## stock_photoyes  1.08031    1.05682   1.022    0.308
## duration       -0.02681    0.19041  -0.141    0.888
## wheels          7.28518    0.55469  13.134  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.901 on 136 degrees of freedom
## Multiple R-squared:  0.719,  Adjusted R-squared:  0.7108
## F-statistic: 87.01 on 4 and 136 DF,  p-value: < 2.2e-16
```

Using this output, we identify the point estimates b_i of each β_i, just as we did in the one-predictor case.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	36.2110	1.5140	23.92	0.0000
cond_new	-5.1306	1.0511	4.88	0.0000
stock_photo	1.0803	1.0568	1.02	0.3085
duration	-0.0268	0.1904	-0.14	0.8882
wheels	7.2852	0.5547	13.13	0.0000
				$df = 136$

Multiple regression model
A multiple regression model is a linear model with many predictors. In general, we write the model as

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k$$

when there are k predictors. We often estimate the β_i parameters using a computer.

Exercise: Write out the the multiple regression model using the point estimates from regression output. How many predictors are there in this model?^[$\hat{y} = 36.21 + -5.13x_1 + 1.08x_2 - 0.03x_3 + 7.29x_4$, and there are $k = 4$ predictor variables.

Exercise:
What does β_4, the coefficient of variable x_4 (Wii wheels), represent? What is the point estimate of β_4?[2]

Example:
Compute the residual of the first observation in the dataframe using the regression equation.

```
mario_mod_multi$residuals[1]
```

```
##        1
## 1.923402
```

The `broom` package has a function `augment` that will calculate the predicted and residuals.

```
library(broom)
```

```
augment(mario_mod_multi) %>%
  head(1)
```

```
## # A tibble: 1 x 11
##   total_pr cond  stock_photo duration wheels .fitted .resid .std.resid   .hat
##      <dbl> <fct> <fct>          <int>  <int>   <dbl>  <dbl>      <dbl>  <dbl>
## 1     51.6 new   yes                3      1    49.6   1.92      0.397 0.0215
## # ... with 2 more variables: .sigma <dbl>, .cooksd <dbl>
```

$e_i = y_i - \hat{y}_i = 51.55 - 49.62 = 1.93$

[2] It is the average difference in auction price for each additional Wii wheel included when holding the other variables constant. The point estimate is $b_4 = 7.29$.

Example: We estimated a coefficient for `cond` as $b_1 = -10.90$ with a standard error of $SE_{b_1} = 1.26$ when using simple linear regression. Why might there be a difference between that estimate and the one in the multiple regression setting?

If we examined the data carefully, we would see that some predictors are correlated. For instance, when we estimated the connection of the outcome `total_pr` and predictor `cond` using simple linear regression, we were unable to control for other variables like the number of Wii wheels included in the auction. That model was biased by the confounding variable `wheels`. When we use both variables, this particular underlying and unintentional bias is reduced or eliminated (though bias from other confounding variables may still remain).

Example: The previous example describes a common issue in multiple regression: correlation among predictor variables. We say the two predictor variables are **collinear** (pronounced as **co-linear**) when they are correlated, and this collinearity complicates model estimation. While it is impossible to prevent collinearity from arising in observational data, experiments are usually designed to prevent predictors from being collinear.

Exercise:
The estimated value of the intercept is 41.34, and one might be tempted to make some interpretation of this coefficient, such as, it is the model's predicted price when each of the variables take value zero: the game is new, the primary image is not a stock photo, the auction duration is zero days, and there are no wheels included. Is there any value gained by making this interpretation?[3]

Adjusted R^2 as a better estimate of explained variance

We first used R^2 in simple linear regression to determine the amount of variability in the response that was explained by the model:

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in the outcome}} = 1 - \frac{Var(e_i)}{Var(y_i)}$$

where e_i represents the residuals of the model and y_i the outcomes. This equation remains valid in the multiple regression framework, but a small enhancement can often be even more informative.

Exercise: The variance of the residuals for the model is 23.34, and the variance of the total price in all the auctions is 83.06. Calculate R^2 for this model.[4]

```
augment(mario_mod_multi) %>%
  summarise(var_resid=var(.resid))
```

```
## # A tibble: 1 x 1
##   var_resid
##       <dbl>
## 1      23.3
```

```
mariokart_new %>%
  summarise(total_var=var(total_pr))
```

[3] Three of the variables (`cond`, `stock_photo`, and `wheels`) do take value 0, but the auction duration is always one or more days. If the auction is not up for any days, then no one can bid on it! That means the total auction price would always be zero for such an auction; the interpretation of the intercept in this setting is not insightful.

[4] $R^2 = 1 - \frac{23.34}{83.06} = 0.719$.

```
## # A tibble: 1 x 1
##   total_var
##       <dbl>
## 1      83.1
```

```
1-23.34/83.05864
```

```
## [1] 0.7189937
```

```
summary(mario_mod_multi)$r.squared
```

```
## [1] 0.7190261
```

This strategy for estimating R^2 is acceptable when there is just a single variable. However, it becomes less helpful when there are many variables. The regular R^2 is actually a biased estimate of the amount of variability explained by the model. To get a better estimate, we use the adjusted R^2.

> Adjusted $\mathbf{R^2}$ as a tool for model assessment
> The adjusted $\mathbf{R^2}$ is computed as
>
> $$R^2_{adj} = 1 - \frac{Var(e_i)/(n-k-1)}{Var(y_i)/(n-1)} = 1 - \frac{Var(e_i)}{Var(y_i)} \times \frac{n-1}{n-k-1}$$
>
> where n is the number of cases used to fit the model and k is the number of predictor variables in the model.

Because k is never negative, the adjusted R^2 will be smaller – often times just a little smaller – than the unadjusted R^2. The reasoning behind the adjusted R^2 lies in the **degrees of freedom** associated with each variance.[5]

> **Exercise**:
> There were $n = 141$ auctions in the `mariokart` data set and $k = 4$ predictor variables in the model. Use n, k, and the appropriate variances to calculate R^2_{adj} for the Mario Kart model.[6]

```
summary(mario_mod_multi)$adj.r.squared
```

```
## [1] 0.7107622
```

> **Exercise*:
> Suppose you added another predictor to the model, but the variance of the errors $Var(e_i)$ didn't go down. What would happen to the R^2? What would happen to the adjusted R^2?^[The unadjusted R^2 would stay the same and the adjusted R^2 would go down.

File Creation Information

- File creation date: 2020-07-31
- Windows version: Windows 10 x64 (build 18362)
- R version 3.6.3 (2020-02-29)
- `mosaic` package version: 1.7.0
- `tidyverse` package version: 1.3.0

[5] In multiple regression, the degrees of freedom associated with the variance of the estimate of the residuals is $n-k-1$, not $n-1$. For instance, if we were to make predictions for new data using our current model, we would find that the unadjusted R^2 is an overly optimistic estimate of the reduction in variance in the response, and using the degrees of freedom in the adjusted R^2 formula helps correct this bias.

[6] $R^2_{adj} = 1 - \frac{23.34}{83.06} \times \frac{141-1}{141-4-1} = 0.711$.