Introduction to Statistical Inference (QTM 100 Lab)

Lecture 11: Linear Regression

Justin Eloriaga — Emory University

Fall 2024

Gameplan

Preliminaries

Correlation

Simple Linear Regression

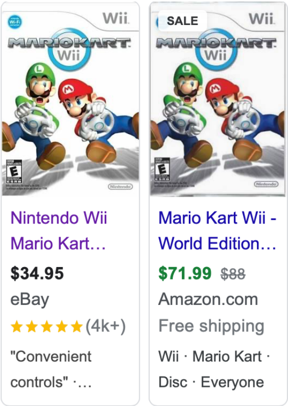
Residuals

Preliminaries

Overview

- We now discuss what is, indeed, the most used statistical tool in establishing linear relationships, the *linear regression*!
- We now actually discuss some things more commonly used in empirical research
  - Bivariate Regression
  - Multivariate Regression

Mario Kart



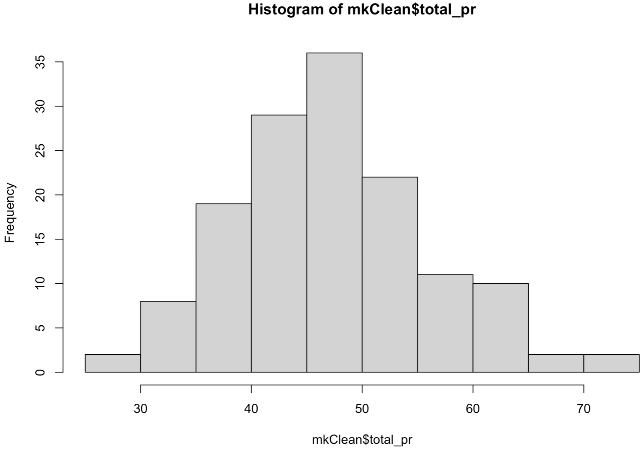
- Remember Mario Kart? I'm sure at least 3/4 of the class played this before.
- We will explore a dataset that includes all auctions on ebay for a full week in October 2009.

Variable	Description
`ID`	Auction ID assigned by Ebay.
`duration`	Auction length, in days.
`n_bids`	Number of bids.
`cond`	Game condition, either new or used.
`start_pr`	Starting price of the auction.
`ship_pr`	Shipping price.
`total_pr`	Total price, which equals the auction price plus the shipping price.
`ship_sp`	Shipping speed or method.
`seller_rate`	The seller's rating on Ebay (number of positive ratings minus the number of negative ratings).
`stock_photo`	Whether or not the auction feature photo was a "stock" photo.
`wheels`	Number of Wii wheels included in the auction.
`title`	The title of the auctions.

Recall mkClean

We will use the cleaned version of the data which excludes the two packages that cost more than 100 dollars. Use a hist() to check

```
mkClean <- subset(mariokart, mariokart$total_pr<100)
```



Correlation

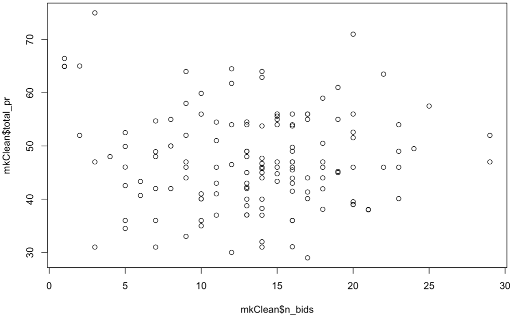
Bids and Prices

Question: Is there a relationship between the total selling price and number of bids the package received?

Bids and Prices

Question: Is there a relationship between the total selling price and number of bids the package received? Let's try to investigate with a scatterplot

```
plot(mkClean$n_bids, mkClean$total_pr)
```



The correlation appears to be just a *random scatter!*

Using `cor()` and `cor.test()`

We can use the `cor()` and `cor.test()` functions to test for the *correlations of these variables more formally*

```
cor(mkClean$n_bids, mkClean$total_pr)
cor.test(mkClean$n_bids, mkClean$total_pr)
```

```
        Pearson's product-moment correlation

data:  mkClean$n_bids and mkClean$total_pr
t = -0.93113, df = 139, p-value = 0.3534
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.24090869  0.08772181
sample estimates:
        cor
-0.07873206
```

At a 95% CI for the true correlation between total price and number of bids, suggests that the true population correlation **may be zero** (since $p = 0.3534$).

Simple Linear Regression

The workflow

We already know that the correlation may not be significant. Nevertheless, to investigate further, we can use a linear regression. This is done in a couple of steps

1. `lm()` Estimating a linear regression model
2. `summary()` Summarizing the results of the model
3. `abline()` Graphing out the regression line
4. `confint()` Getting the confidence intervals for β_0 and β_1
5. `resid()` Checking the residuals
6. `predict()` Getting predicted values

Building the Model

Suppose you want to run the model below

$$total\_pr_i = \beta_0 + \beta_1 n\_bids_i + u_i \text{ where } u_i \sim N(\mu, \sigma^2)$$

In here, y (the dependent variable) is `total_pr` and x (the independent variable) is `n_bids`. We let u_i be our error term. Running this in `R` involves the use of the `lm()` function

```
m1 <- lm(mkClean$mkClean$total_pr ~ mkClean$n_bids)
summary(m1)
```

```
Call:
lm(formula = mkClean$total_pr ~ mkClean$n_bids)

Residuals:
     Min       1Q   Median       3Q      Max
-18.0016  -6.2265  -0.8672   6.5104  26.2756

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     49.0979     1.9470  25.217   <2e-16 ***
mkClean$n_bids  -0.1245     0.1337  -0.931    0.353
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.118 on 139 degrees of freedom
Multiple R-squared:  0.006199,	Adjusted R-squared:  -0.0009509
F-statistic: 0.867 on 1 and 139 DF,  p-value: 0.3534
```

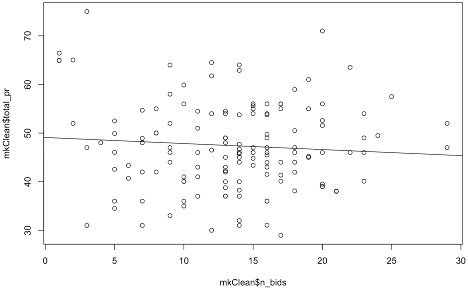
Interpreting the Model

```
Call:
lm(formula = mkClean$total_pr ~ mkClean$n_bids)

Residuals:
     Min       1Q   Median       3Q      Max 
-18.0016  -6.2265  -0.8672   6.5104  26.2756 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     49.0979     1.9470  25.217   <2e-16 ***
mkClean$n_bids  -0.1245     0.1337  -0.931    0.353    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.118 on 139 degrees of freedom
Multiple R-squared:  0.006199,	Adjusted R-squared:  -0.0009509 
F-statistic: 0.867 on 1 and 139 DF,  p-value: 0.3534
```

- The model estimates the regression line $\hat{y} = 49.1 - 0.12\texttt{n\_bids}$.
- With zero bids, the predicted `total_pr` of a package is $49.1
- For every additional bid the package receives, the total price *decreases* by roughly 12 cents
- Intercept is *significant* but the slope is *not significant*

Adding the Regression Line

One can use the `abline()` command to add the regression line to any scatterplot



Clearly, the line is downward sloping. The intercept is roughly at 49.1 with a slope of -0.12. Those are the things we estimated from the linear regression.

Confidence Intervals

It is important to get a sense of the inference by using confidence intervals. Let's get the confidence intervals for $\hat{\beta}_0$ (i.e. the intercept estimate) and $\hat{\beta}_1$ (i.e. the slope estimate). To do this, we use the `confint()` function

```
> confint(m1)
                     2.5 %     97.5 %
(Intercept)     45.2482781 52.9475346
mkClean$n_bids  -0.3888219  0.1398502
```

Clearly, the insignificance is seen for the slope coefficient since 0 is in the confidence interval.

Residuals

Residuals

There are 2 ways produce residuals for each observation in the dataset.

1. You can obtain the *regular residuals*
2. You can obtain the *standardized residuals*

$$\text{Standardized } \hat{u}_i = \frac{\hat{u}_i}{\hat{\sigma}_u}$$

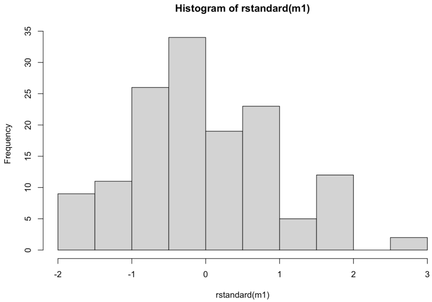
We use the following lines of code to get the regular residuals and standardized residuals

```
m1$residuals
resid(m1)
rstandard(m1)
```

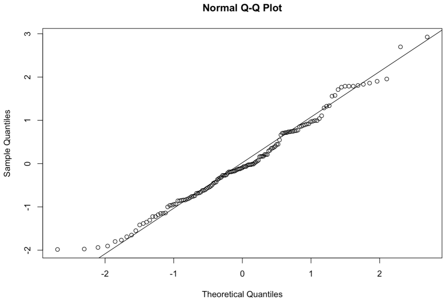
You can also get the predicted values $\hat{y}$ using the `predict()` command.

Visualizing Residuals

Let's use a combination of `hist()`, `qqnorm()`, and `qqline()` plots.



`hist(rstandard(m1))`



`qqnorm(rstandard(m1))`
`qqline(rstandard(m1))`

Are the residuals *normally distributed*?

Zero in on Row 1

We know that the following relationship must hold

$$y_i = \hat{y}_i + \hat{u}_i$$

That is, the true value of y is equal to your estimated value in addition to some error. Let's zero in on row one to make sure this is the case. Run `mkClean[1,]`, `predict(m1)[1]`, `resid(m1)[1]` to specifically pull the results for the first observation.

```
> mkClean[1,]
            id duration n_bids cond start_pr ship_pr total_pr  ship_sp seller_rate stock_photo wheels
1 150377422259        3     20  new     0.99       4    51.55 standard        1580         yes      1
                                                          title
1 ~~ Wii MARIO KART &amp; WHEEL ~ NINTENDO Wii ~ BRAND NEW ~~
> predict(m1)[1]
       1
46.60819
> resid(m1)[1]
       1
4.941811
> 46.60819 + 4.941811
[1] 51.55
```

Assessing More Assumptions

We have assumptions on *linearity* and having a *constant* variance.

- We can compare the residuals to the fitted values
- Plot residuals vs. fitted values, then, place a line at 0.

```
plot(predict(m1),rstandard(m1),xlab = "Fitted
Values", ylab = "Standardized Residuals")
abline(h= 0, lty = 2)
```