

Log-Transforming the Outcome

2020-01-06

Preparation

In this set of notes, you will learn another method of dealing with nonlinearity and also for dealing with heterogeneity of variance. Specifically, we will look at log-transforming the predictor in a linear model. To do so, we will use the *movies.csv* dataset (see the [data codebook](#)) to explain variation in movie budgets.

```
# Load libraries
library(broom)
library(corr)
library(educate) #Need version 0.1.0.1
library(patchwork)
library(tidyverse)

# Read in data
movies = read_csv(file = "~/Documents/github/epsy-8252/data/movies.csv")
head(movies)
```

```
# A tibble: 6 x 6
  title      budget year  age genre length
  <chr>      <dbl> <dbl> <dbl> <chr>  <dbl>
1 Trapped    41.9   2002   16 Other    106
2 Urbania     0.328  2000   18 Drama    106
3 End of Violence, The 7.82   1997   21 Drama    122
4 Snatch.    14.6   2000   18 Comedy   102
5 Narc      10.5   2002   16 Drama    105
6 Kissing Jessica Stein 1.42   2001   17 Comedy    97
```

Relationship between Budget and Running Time

To begin the analysis, we will examine the marginal distributions of movie running time (predictor) and budget (outcome), as well as a scatterplot between them for our sample data. We place the plots side-by-side using functionality from the `patchwork` package. (See the [documentation for this package here](#).)

```
# Marginal distribution of budget (outcome)
p1 = ggplot(data = movies, aes(x = budget)) +
  stat_density() +
  theme_bw() +
  xlab("Movie budget (in millions of dollars)") +
  ylab("Probability density")

# Marginal distribution of running time (predictor)
p2 = ggplot(data = movies, aes(x = length)) +
  stat_density() +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Probability density")

# Scatterplot
p3 = ggplot(data = movies, aes(x = length, y = budget)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Movie budget (in millions of dollars)") +
  annotate(geom = "text", x = 90, y = 679, label = "The Time Machine (1960)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 114, y = 542, label = "Sabrina (1960)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 108, y = 443, label = "The Nutty Professor (1963)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 181, y = 23, label = "1776 (1972)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 187, y = 92, label = "The Green Mile (1999)", size = 3, hjust = 1)

# Place figures side-by-side
p1 + p2 + p3
```

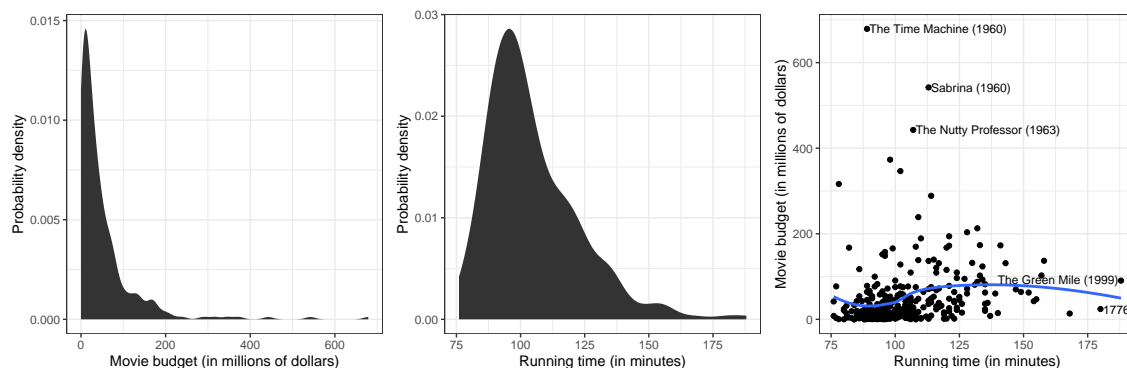


Figure 1. Density plot of the distribution of movie budgets (left) and running time (center). The scatterplot (right) shows the relationship between running time and budget. The loess smoother (blue, dashed line) is also displayed on the scatterplot. All movies that have a budget over 400 million dollars or a running time of at least three hours are also identified.

The distribution of movie budgets is severely right-skewed. Although the median budget is around 25 million dollars, the plot shows evidence that several movies have exorbitant budgets. There is also one movie (*The Time Machine*) that has a budget of close to 680 million dollars; a budget that is over eight standard deviations higher than the median budget. The distribution of running time also appears right-skewed, with most movies clocking in at around 100 minutes. There are two movies in the sample that have a running time near three hours.

The scatterplot suggests a weak, slightly positive relationship between running time of a movie and budget. It also suggests several potential problems with fitting a linear model to the data:

- The relationship seems slightly curvilinear.
- The variation in budget for shorter movies is greater than the variation in budget for longer movies (heteroskedasticity).
- There are more data on the lower half of the plot than the upper half. This posits a potential problem with the normality assumption.

We can see these assumption violations much more clearly in the scatterplot of residuals versus fitted values we get from regressing movie budgets on running time.

```
# Fit model
lm.1 = lm(budget ~ 1 + length, data = movies)

# Obtain residuals and fitted values
out_1 = augment(lm.1)

# Density plot of the residuals
p1 = ggplot(data = out_1, aes(x = .std.resid)) +
  stat_density_confidence(model = "normal") +
  stat_density(geom = "line") +
  theme_bw() +
  xlab("Standardized residuals") +
  ylab("Probability density")

# Residuals versus fitted values
p2 = ggplot(data = out_1, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth() +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals")

# Plot the figures side-by-side
p1 + p2
```

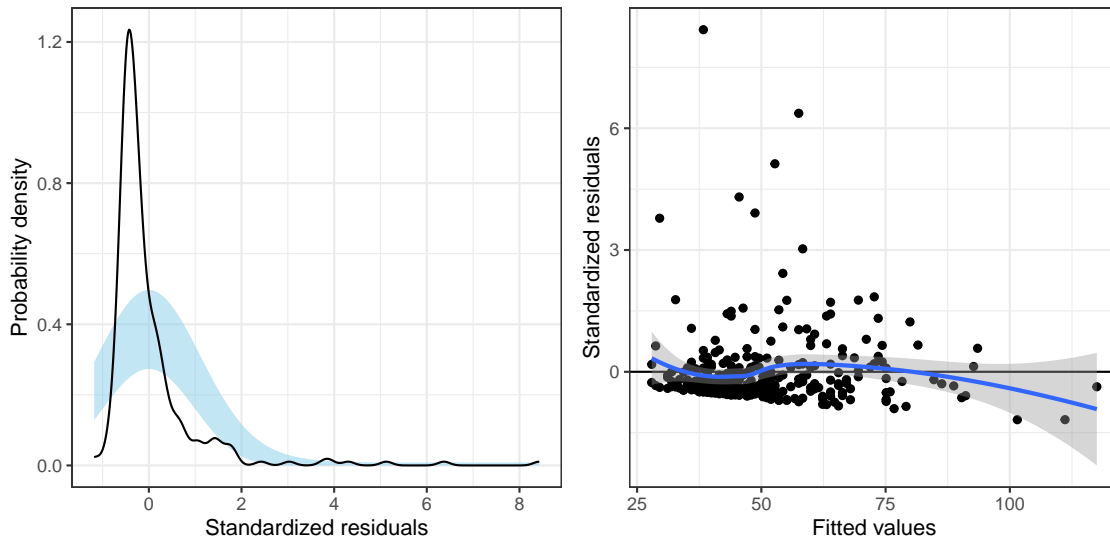


Figure 2. Density plot of the standardized residuals (left) and scatterplot of the standardized residuals versus the fitted values (right) from regressing budget on running time. The confidence envelope from assuming a normal distribution is also displayed along with the density plot. In the scatterplot, the loess smoother (solid, blue line) is displayed along with the 95% confidence envelope (grey shaded area) and the reference line of $Y=0$.

These plots suggest violations of the normality assumption (the marginal distribution of the residuals is right-skewed) and of the assumption of homoskedasticity. The assumption that the average residual is 0, may be reasonable on the basis of this plot.

Transform the Outcome Using the Natural Logarithm (Base-e)

To alleviate problems of non-normality when the conditional distributiona of the outcome are right-skewed (or have high-end outliers) OR to alleviate heteroskedasticity, we can mathematically transform the outcome using a logarithm. Any base can be used for the logarithm, but we will transform the outcome using the natural logarithm because of the interpretive value.

First, we will create the log-transformed budget as a new column in the data, and then we will use the log-transformed budget (rather than raw budget) in any analyses.

```
# Create log-transformed budget
movies = movies %>%
  mutate(
    Lbudget = log(budget)
  )

# Examine data
head(movies)
```

```
# A tibble: 6 x 7
  title      budget year  age genre length Lbudget
  <chr>      <dbl> <dbl> <dbl> <chr>  <dbl>  <dbl>
1 Trapped    41.9   2002   16 Other    106    3.73
2 Urbania     0.328  2000   18 Drama     106   -1.11
3 End of Violence, The  7.82   1997   21 Drama     122    2.06
```

4	Snatch.	14.6	2000	18	Comedy	102	2.68
5	Narc	10.5	2002	16	Drama	105	2.35
6	Kissing Jessica Stein	1.42	2001	17	Comedy	97	0.350

Recall that the logarithm is the inverse function of an exponent. As an example, consider the budget and natural log-transformed budget for *Trapped*.

$$\ln(41.88) = 3.73$$

Remember, in this case, the logarithm answers the mathematical question:

e to what power is equal to 41.88?

Answering this,

$$e^{3.73} = 41.88$$

Re-analyze using the Log-Transformed Budget

Now we will re-examine the scatterplot using the log-transformed outcome to see how this transformation affects the relationship.

```
# Scatterplot
ggplot(data = movies, aes(x = length, y = Lbudget)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Movie budget (in millions of dollars)") +
  annotate(geom = "text", x = 90, y = 6.53, label = "The Time Machine (1960)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 114, y = 6.31, label = "Sabrina (1960)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 108, y = 6.10, label = "The Nutty Professor (1963)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 181, y = 3.17, label = "1776 (1972)", size = 3, hjust = 0) +
  annotate(geom = "text", x = 187, y = 4.51, label = "The Green Mile (1999)", size = 3, hjust = 1)
```

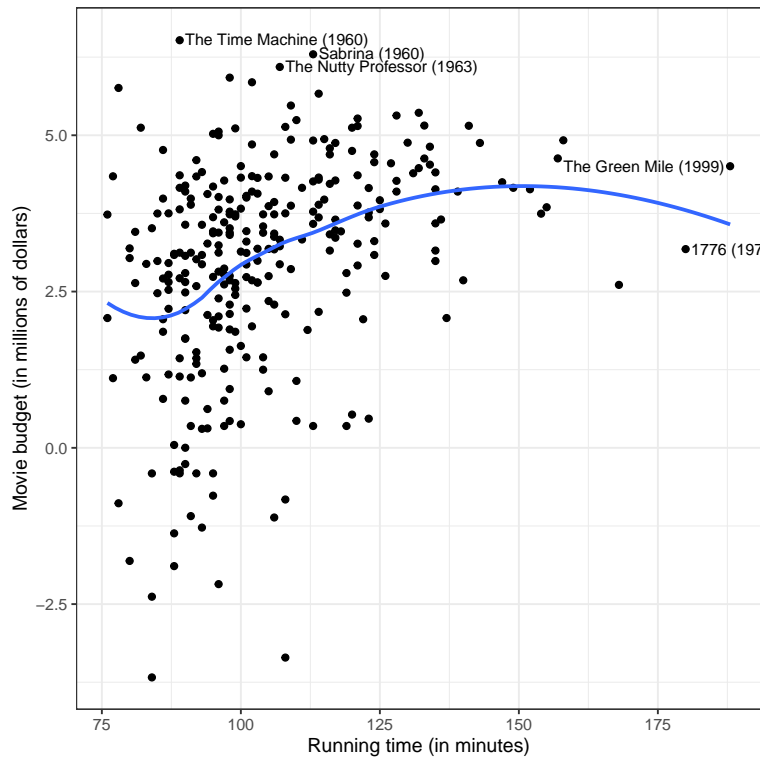


Figure 3. Scatterplot between age and log-transformed budget for the sample of movies. The loess smoother (blue, dashed line) is also displayed.

Log-transforming the outcome has drastically affected the scale for the outcome. The three movies which had extremely high budgets when we examined raw budget, no longer seems like outliers in the transformed data. Has this helped us better meet the assumptions? Again, we should examine the residual plots.

```
# Fit model
lm.2 = lm(Lbudget ~ 1 + length, data = movies)

# Obtain residuals and fitted values
out_2 = augment(lm.2)
```

```
# Density plot of the residuals
p1 = ggplot(data = out_2, aes(x = .std.resid)) +
  stat_density_confidence(model = "normal") +
  stat_density(geom = "line") +
  theme_bw() +
  xlab("Standardized residuals") +
  ylab("Probability density")

# Residuals versus fitted values
p2 = ggplot(data = out_2, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth() +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals")

# Plot the figures side-by-side
p1 + p2
```

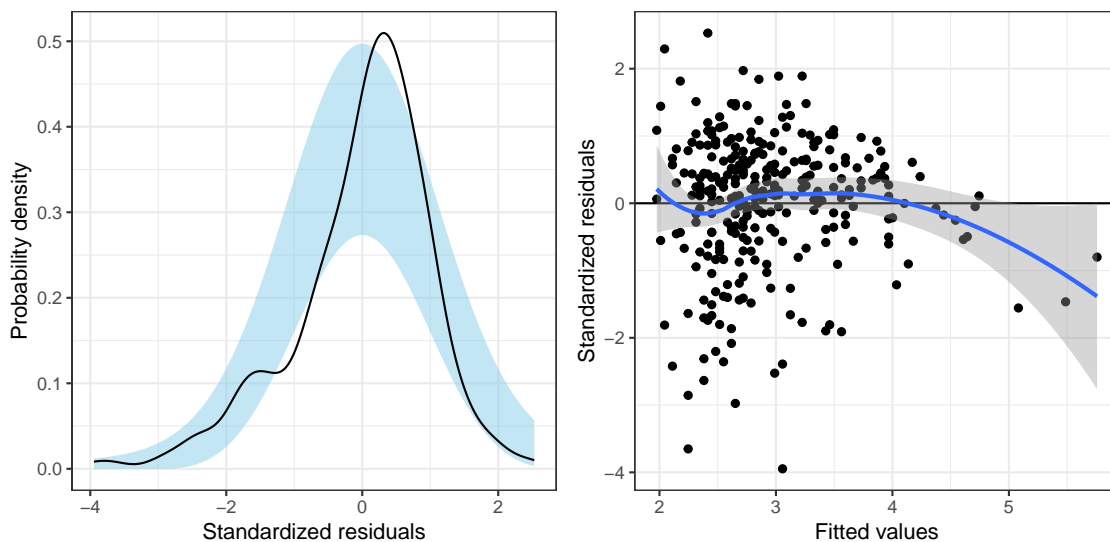


Figure 4. Density plot of the standardized residuals (left) and scatterplot of the standardized residuals versus the fitted values (right) from regressing budget on age. The confidence envelope from assuming a normal distribution is also displayed along with the density plot. In the scatterplot, the loess smoother (solid, blue line) is displayed along with the 95% confidence envelope (grey shaded area) and the reference line of $Y=0$.

These plots suggest that after the transformation, there is a great deal of improvement in meeting the assumption of normality, however there looks to still be some violation. The assumption of homoskedasticity now shows only minor violation, having improved over residuals from the non-transformed budget. (The biggest problems seems to be for movies with longer running times for which there are fewer observations; thus it might be a few observations driving this violation.) The assumption that the average residual is 0 now shows some violation, which, again., may be driven by those movies with longer running times.

Interpreting the Regression Output

Let's examine the output from the model in which we regressed the log-transformed budget on running time.

```
# Model-level output  
glance(lm.2)
```

```
# A tibble: 1 x 11  
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC  
    <dbl>      <dbl> <dbl>    <dbl>   <dbl> <int> <dbl> <dbl> <dbl>  
1     0.124      0.121  1.63     41.2 5.58e-10     2  -556. 1117. 1128.  
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

The model-level summary information suggests that differences in running time explains 12.4% of the variation in budgets. (Remember, explaining variation in log-budget is the same as explaining variation in budget). Although this is a small amount of explained variation, it is more than we expect because of sampling variation, $F(1, 290) = 41.21$, $p < 0.001$.

```
# Coefficient-level output  
tidy(lm.2)
```

```
# A tibble: 2 x 5  
  term          estimate std.error statistic p.value  
  <chr>         <dbl>    <dbl>    <dbl>   <dbl>  
1 (Intercept)  -0.589     0.558     -1.05 2.93e- 1  
2 length       0.0338    0.00526     6.42 5.58e-10
```

From the coefficient-level output, the fitted equation is:

$$\ln(\widehat{\text{Budget}}_i) = -0.59 + 0.034(\text{Running Time}_i)$$

With log-transformations, there are two possible interpretations we can offer. The first is to interpret the coefficients using the log-transformed metric. These we interpret in the exact same way we do any other regression coefficients (except we use log-outcome instead of outcome):

- The intercept, $\hat{\beta}_0 = -0.59$, is the average predicted log-budget for movies with a running time of 0 minutes (extrapolation).
- The slope, $\hat{\beta}_1 = 0.034$, indicates that each one-minute difference in running time is associated with a log-budget that differs by 0.034, on average.

Back-Transforming: A More Useful Interpretation

A second, probably more useful, interpretation is to back-transform the metric of log-budget to the metric of raw budget. To think about how to do this, we first consider a more general expression of the fitted linear model:

$$\ln(\hat{Y}_i) = \hat{\beta}_0 + \hat{\beta}_1(X_i)$$

The left-hand side of the equation is in the log-transformed metric, which drives our interpretations. If we want to instead, interpret using the raw metric of Y , we need to back-transform from $\ln(Y)$ to Y . To back-transform, we use the inverse function, which is to exponentiate using the base of the logarithm, in our case, base- e .

$$e^{\ln(Y_i)} = Y_i$$

If we exponentiate the left-hand side of the equation, to maintain the equality, we also need to exponentiate the right-hand side of the equation.

$$e^{\ln(Y_i)} = e^{\hat{\beta}_0 + \hat{\beta}_1(X_i)}$$

Then we use rules of exponents to simplify this.

$$Y_i = e^{\hat{\beta}_0} \times e^{\hat{\beta}_1(X_i)}$$

For our example, when we exponentiate both sides of the fitted equation:

$$\hat{\text{Budget}}_i = e^{-0.59} \times e^{0.034(\text{Running Time}_i)}$$

Substituting in Values for Age to Interpret Effects

To interpret the effects, which are now interpreted using the metric of budget (not log-budget), we can substitute in the different values for running time and solve. For example when running time = 0:

$$\begin{aligned}\hat{\text{Budget}}_i &= e^{-0.59} \times e^{0.034(0)} \\ &= 0.56 \times 1 \\ &= 0.56\end{aligned}$$

The predicted budget for a movie that is 0 minutes long is 0.56 million dollars (extrapolation). How about a movie that is one minute long (a one-minute difference)?

$$\begin{aligned}\hat{\text{Budget}}_i &= e^{-0.59} \times e^{0.034(1)} \\ &= 0.56 \times 1.034 \\ &= 0.57\end{aligned}$$

The predicted budget for a movie that is one minute long is 0.57 million dollars. This is 1.034 TIMES the budget of a movie that is 0 minutes long. Rather than using the language of **TIMES difference** you could also use the language of **fold difference**. In this case the slope coefficient would be interpreted as,

Each one-minute difference in running time is associated with a 1.034-fold difference in budget, on average.

Simply put, when we back-transform from interpretations of log- Y to Y the interpretations are multiplicatively related to the intercept rather than additively related. We can obtain these multiplicative values (and the back-transformed intercept) by using the `exp()` function to exponentiate the coefficients from the fitted model, which we obtain using the `coef()` function.

```
exp(coef(lm.2))
```

```
(Intercept)      length
  0.5550024    1.0343293
```

Approximate Interpretation of the Slope as Percent Change

Remember from the previous unit that by using the natural logarithm we can interpret the effects as *percent change*. Rather than saying that each one-minute difference in running time is associated with a 1.034-fold difference in budget, on average, we can directly interpret the slope as the percent change. Thus $\hat{\beta}_1 = 0.034$ can be interpreted as:

Each one-minute difference in running time is associated with a 3.4 percent increase in budget, on average.

This is an approximation that is often “good enough” for low values of the slope (e.g., $\hat{\beta}_k < 0.20$).

If you use the language of percent change, be very careful. *Percent* and *Percent change* typically indicate differing amounts! For instance, consider two movie budgets, one is 20 million dollars and the other is 15 million dollars. The movie with the 15 million dollar budget has a budget that is 75% of the movie that has a 20 million dollar budget. However, the percent change in budgets is 25%.

Plotting the Fitted Model

As always, we should plot the fitted model to aid in interpretation. To do this we will use this expression of the fitted equation:

$$\widehat{\text{Budget}}_i = e^{-0.59} \times e^{0.034(\text{Running Time}_i)}$$

This can be added to the `stat_function()` layer of `ggplot()`.

```
# Plot
ggplot(data = movies, aes(x = length, y = budget)) +
  geom_point(alpha = 0) +
  stat_function(fun = function(x) {exp(-0.59) * exp(0.034*x)} ) +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Predicted budget (in millions of dollars)")
```

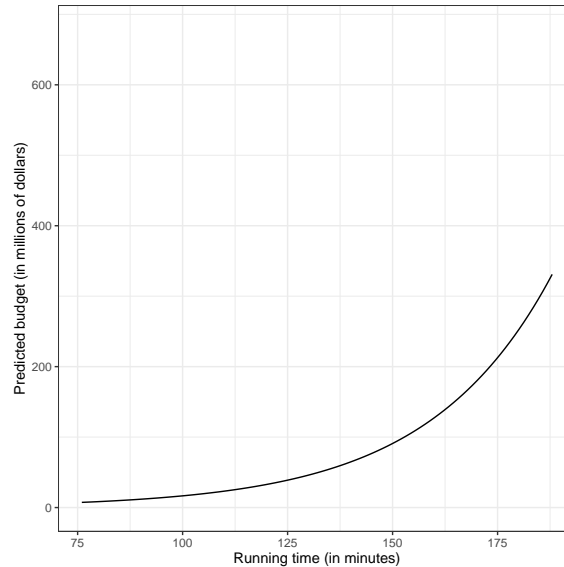


Figure 5. Plot of the predicted movie budget as a function of its running time.

Based on this plot, we see a non-linear, positive effect of running time on budget. Shorter movies tend to have a smaller budget, on average, but the increase in budget as movies get longer is not constant. This pattern of non-linear increase is referred to as *positive exponential growth* (it has a positive increasing rate of growth). This pattern is different than what we saw in the fitted equation for the college data from the previous unit, which could be described as *positive exponential decay* (positive diminishing rates of growth). However, both of these effects are described by *monotonic* functions which have no change in the direction; both were always increasing (positive).

Modeling the Remaining Non-Linearity

How do we model the non-linearity we observed in the relationship even after we log-transformed the outcome? The pattern in the scatterplot shown by the Loess smoother in Figure 3, suggests a curvilinear pattern similar to what we observed in the college data in Unit 2 and 3. This means we can try to model the non-linearity by (1) log-transforming the predictor (in addition to log-transforming the outcome), or (2) include a polynomial effect of running time in the model. How do you choose? Fit both models and examine the residuals.

```
# Fit log-log model
lm_log = lm(Lbudget ~ 1 + log(length), data = movies)

# Fit polynomial model
lm_poly = lm(Lbudget ~ 1 + length + I(length^2), data = movies)

# Obtain residuals
out_log = augment(lm_log)
out_poly = augment(lm_poly)
```

```
# Log-log residuals
p1 = ggplot(data = out_log, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth() +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals") +
  ggtitle("Log-Log Model")

# Polynomial model residuals
p2 = ggplot(data = out_poly, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth() +
  theme_bw() +
  xlab("Fitted values") +
  ylab("Standardized residuals") +
  ggtitle("Polynomial Model")

# Plot side-by-side
p1 + p2
```

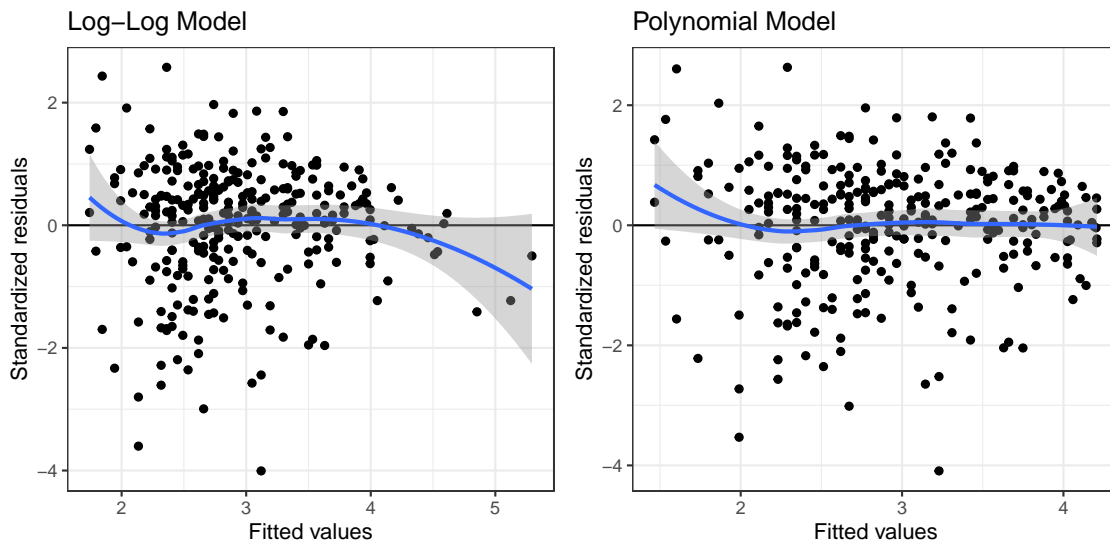


Figure 6. Standardized residuals versus the fitted values for the model using the natural logarithm of running time (left) and for the model that includes a linear and quadratic effect of running time (right) to predict variation in log-budget. The loess smoother and confidence envelope for the smoother is also included in both plots.

Based on the plots, the polynomial model seems to show better fit to the assumption that the average residual is 0 than the log-log model. We can also see this by comparing the fit of each model to the pattern modeled by the loess smoother in the scatterplot of the log-budgets versus the running times. To do this we would need to obtain the coefficient-level output for each fitted model and then use `stat_function()` to include the fitted curve onto the scatterplot.

```

# Log-log model
p1 = ggplot(data = movies, aes(x = length, y = Lbudget)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  stat_function(fun = function(x) {log(exp(-15.2) * x^(3.92))},
    color = "red", linetype = "dashed") +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Movie budget (in millions of dollars)") +
  ggtitle("Log-Log Model")

# polynomial model
p2 = ggplot(data = movies, aes(x = length, y = Lbudget)) +
  geom_point() +
  geom_smooth(se = FALSE) +
  stat_function(fun = function(x) {-6.28 + 0.135*x - 0.000434*x^2},
    color = "red", linetype = "dashed") +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Movie budget (in millions of dollars)")

# Plot side-by-side
p1 + p2

```

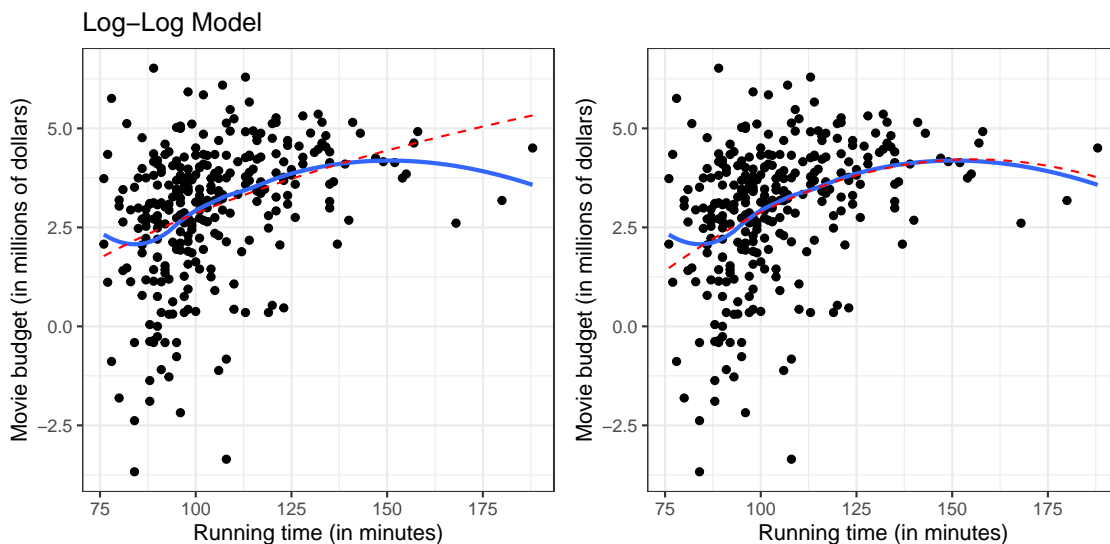


Figure 7. Scatterplot between age and log-transformed budget for the sample of movies. The loess smoother (blue, dashed line) is also displayed. The fitted model using the natural logarithm of running time (left) and for the model that includes a linear and quadratic effect of running time (right) to predict variation in log-budget is also included (red, dashed line)

The polynomial model shows better adherence to the pattern captured by the loess smoother than the log-log model. Thus, we will adopt the polynomial model over the log-log model.

```
# Model-level output
glance(lm_poly)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
    <dbl>      <dbl> <dbl>    <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>
1    0.142        0.137  1.61     24.0 2.27e-10     3 -553. 1113. 1128.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

The model-level summary information suggests that the model explains 14.2% of the variation in budgets. This is, again, more than we expect because of sampling variation, $F(2, 289) = 24.00$, $p < 0.001$.

```
# Coefficient-level output
tidy(lm_poly)
```

```
# A tibble: 3 x 5
  term          estimate std.error statistic p.value
  <chr>         <dbl>    <dbl>    <dbl>   <dbl>
1 (Intercept) -6.28      2.37     -2.65 0.00860
2 length       0.135    0.0413     3.26 0.00124
3 I(length^2) -0.000434 0.000176    -2.47 0.0143
```

From the coefficient-level output, the fitted equation is:

$$\ln(\widehat{\text{Budget}}_i) = -6.28 + 0.135(\text{Running Time}_i) - 0.0004(\text{Running Time}_i^2)$$

Since the quadratic term is adopted, $t(289) = -2.47$, $p = 0.014$, we interpret the quadratic effect only as an interaction. Namely that:

The effect of running time on log-budget varies by running time.

To better understand the nature of this relationship we plot the fitted curve (see Figure 7). This suggests that there is a curvilinear relationship between running time and log-budget. In general, the effect of running time on log-budget increases for movies with longer running times until about 170 minutes (approximate x -coordinate of the vertex) then the effect of running time has an increasing negative effect on log-budget.

Interpretation in the Raw Budget Metric

We can also back-transform the metric of log-budget to the metric of raw budget. Plotting this function will allow us to understand the relationship between running time and budget from the polynomial model. Exponentiating both sides of the fitted equation:

$$\widehat{\text{Budget}}_i = e^{-6.28} \times e^{0.135(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)}$$

Inputting this into the `stat_function()` layer of our `ggplot()` syntax, we get:

```
# Plot
ggplot(data = movies, aes(x = length, y = budget)) +
  geom_point(alpha = 0) +
  stat_function(fun = function(x) {exp(-6.28) * exp(0.135*x) * exp(-0.0004*x^2)} ) +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Predicted budget (in millions of dollars)")
```

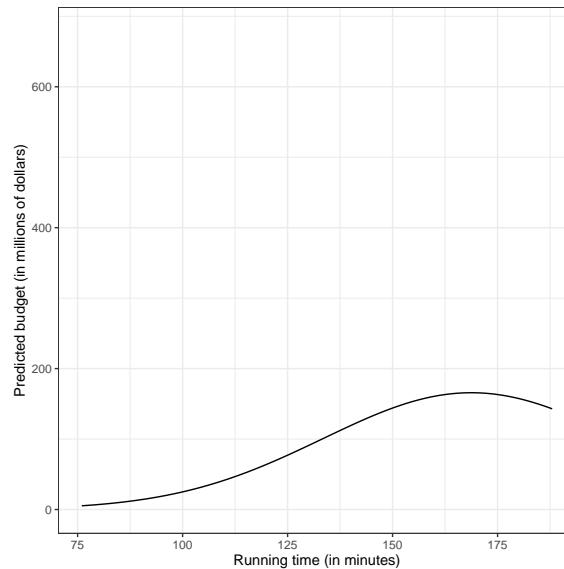


Figure 8. Plot of the predicted movie budget as a function of its running time for the polynomial model.

The relationship between running time and budget is quite complicated. It combines the positive exponential growth we saw earlier with the change in curvature of the negative quadratic effect. In general longer movies have an exponentially increasing budget that begins to diminish for movies longer than approximately 170 minutes.

Relationship between Genre and Budget

One hypothesis might be that action movies tend to have higher budgets due to the special effects they include. We will explore this hypothesis to see whether there are differences in budget between action movies and other movie genres. Since we log-transformed budget (the outcome) in the previous analysis we will need to use the log-transformed outcome in this exploration as well.

```
# Create dummy variable
movies = movies %>%
  mutate(
    action = if_else(genre == "Action", 1, 0)
  )
```

```
# Plot the observed data
ggplot(data = movies, aes(x = action, y = Lbudget)) +
  geom_jitter(alpha = 0.2) +
  stat_summary(fun.y = 'mean', geom = "point", size = 4, color = "darkred") +
  theme_bw() +
  xlab("Genre") +
  ylab("ln(Movie Budget)")
```

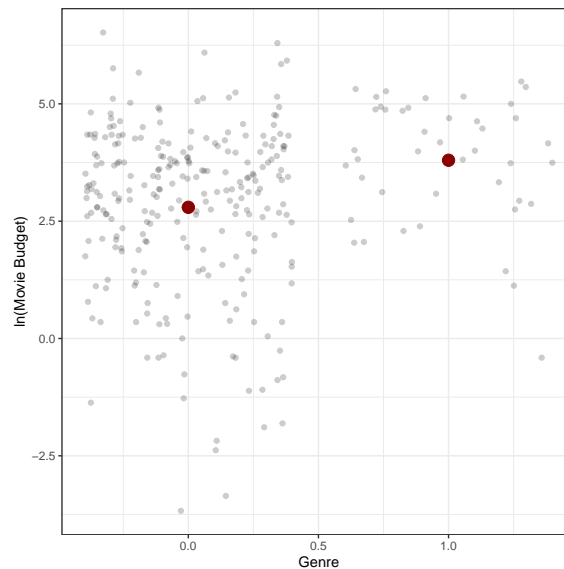


Figure 9. Jittered scatterplot of log-budget versus MPAA rating.

```
# Compute summary statistics
movies %>%
  group_by(action) %>%
  summarize(
    M = mean(Lbudget),
    SD = sd(Lbudget)
  )
```

```
# A tibble: 2 x 3
  action     M     SD
  <dbl> <dbl> <dbl>
1     0  2.80  1.76
2     1  3.80  1.31
```

The jittered scatterplot and summary statistics indicate that the log-budget for action movies is higher than for other genres in our sample. The variation in log-budgets seems roughly comparable between the two groups of movies, although it could be argued that movies in the action genre seem to have slightly less variation than the other genres.

Regressing the Log-Transformed Budget on the Action Dummy

To examine whether the sample differences are more than we would expect because of sampling error, we will fit a model regressing the log-transformed budgets on the action dummy variable.


```
# Fit the model (non-action is reference group)
lm.3 = lm(Lbudget ~ 1 + action, data = movies)
```

```
# Model-level output
glance(lm.3)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
    <dbl>      <dbl> <dbl>    <dbl>  <dbl> <int>  <dbl> <dbl> <dbl>
1  0.0420      0.0387  1.70     12.7 4.24e-4     2 -569. 1143. 1154.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

The model-level summary information suggests that differences in genre (action vs non-action) explains 4.2% of the variation in budget. (Remember, explaining variation in log-budget is the same as explaining variation in budget). Although this is a small amount of variation, it is more than we expect because of sampling variation, $F(1, 290) = 12.71, p = 0.0004$.

```
# Coefficient-level output
tidy(lm.3)
```

```
# A tibble: 2 x 5
  term      estimate std.error statistic  p.value
  <chr>      <dbl>    <dbl>    <dbl>  <dbl>
1 (Intercept)  2.80     0.108     25.9 1.87e-77
2 action       1.00     0.281      3.57 4.24e- 4
```

From the coefficient-level output we can write the fitted equation:

$$\ln(\widehat{\text{Budget}}_i) = 2.80 + 1.003(\text{Action}_i)$$

Coefficient Interpretation: Metric of Log-Budget

With log-transformations, there are several possible interpretations we can offer. The first is to interpret the coefficients using the log-transformed metric. These we interpret in the exact same way we do any other regression coefficients (except we use log-outcome instead of outcome):

- The intercept, $\hat{\beta}_0 = 2.80$, is the average predicted log-budget for non-action movies.
- The slope associated with action movies, $\hat{\beta}_1 = 1.003$, indicates that action movies have a log-budget that is 1.003-units higher than non-action movies, on average.

Coefficient Interpretation: Metric of Budget

We can also interpret the coefficients using the metric of raw budget by back-transforming the coefficients. Remember, if we want to interpret using the raw metric of Y , we need to back-transform from $\ln(Y)$ to Y . To back-transform, we use the inverse function, which is to exponentiate using the base of the logarithm, in our case, base- e . For our example, when we exponentiate both sides of the fitted equation, we get:

$$e^{\ln(\widehat{\text{Budget}}_i)} = e^{2.80 + 1.003(\text{Action}_i)}$$

$$\widehat{\text{Budget}}_i = e^{2.80} \times e^{1.003(\text{Action}_i)}$$

To interpret the effects (which are now interpreted using the metric of raw budget, we can substitute in the different dummy variable patterns and solve.

Non-Action Movie

$$\begin{aligned}\widehat{\text{Budget}}_i &= e^{2.80} \times e^{1.003(0)} \\ &= 16.37 \times 1 \\ &= 16.37\end{aligned}$$

Action Movie

$$\begin{aligned}\widehat{\text{Budget}}_i &= e^{2.80} \times e^{1.003(1)} \\ &= 16.37 \times 2.73 \\ &= 44.61\end{aligned}$$

When we interpret the effects of each of the coefficients,

- Non-action movies have a budget of 16.37 million dollars, on average.
- Action movies have a budget that is 2.73 TIMES the estimated budget for non-action movies, on average.

We can obtain these values directly by exponentiating the estimated regression coefficients.

```
exp(coef(lm.3))
```

(Intercept)	action
16.371205	2.725324

Coefficient Interpretation: Percent Change

Unfortunately, the direct interpretation of the slope coefficients using the language of “percent change” is not trustworthy given that the absolute values of each of the estimated slope is bigger than 0.20. (This direct interpretation from the slope coefficients starts to become untrustworthy when the slope value is higher than about 0.20 or so.)

```
coef(lm.3)
```

(Intercept)	action
2.795524	1.002587

If we did interpret the slope directly, the (wrong) interpretation would be:

Action movies are associated with a 100.2 percent increase in budget, on average, over non-action movies.

If you want the specific percent change in budget, compute

$$\left| 1 - e^{\hat{\beta}_k} \right|$$

This can then be interpreted as a percentage increase or decrease depending on the sign of the slope coefficient.

```
# Percent change
abs(1 - exp(1.002587))
```

```
[1] 1.725323
```

Action movies are associated with a 173 percent increase in budget, on average, over non-action movies.

Again, pay attention to the idea of “percent change” versus “percent”. In this case action movies have a budget which is 273% of non-action movies budgets, on average. This is different than the 173 percent increase in budget that we just interpreted.

Modeling the Effects of Running Time and Genre

We can now explore a final model that includes the main-effects of running time and whether or not the movie is an action movie.

```
# Fit the model (action is reference group)
lm.4 = lm(lbudget ~ 1 + length + I(length^2) + action, data = movies)

# Model-level output
glance(lm.4)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC
  <dbl>      <dbl> <dbl>    <dbl>   <dbl> <int> <dbl> <dbl> <dbl>
1    0.169      0.161  1.59     19.6 1.45e-11     4 -548. 1106. 1124.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

The model-level summary information suggests that the model explains 16.9% of the variation in budget. This is more than we expect because of sampling variation, $F(1, 288) = 19.55, p < .001$.

```
# Coefficient-level output
tidy(lm.4)
```

```
# A tibble: 4 x 5
  term          estimate std.error statistic p.value
  <chr>         <dbl>    <dbl>    <dbl>   <dbl>
1 (Intercept) -5.56      2.35     -2.37 0.0187
2 length       0.121    0.0410     2.95 0.00339
3 I(length^2) -0.000380 0.000174    -2.18 0.0298
4 action        0.807    0.265     3.04 0.00255
```

From the coefficient-level output we can write the fitted equation:

$$\ln(\widehat{\text{Budget}}_i) = -5.56 + 0.12(\text{Running Time}_i) - 0.0004(\text{Running Time}_i^2) + 0.81(\text{Action}_i)$$

Coefficient Interpretation: Metric of Budget

Back-transforming to the metric of budget, the fitted equation is:

$$\widehat{\text{Budget}}_i = e^{-5.56} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)} \times e^{0.81(\text{Action}_i)}$$

Interpreting the effects of running time and action:

- After controlling for whether or not a movie is an action movie, the effect of running time on budget varies by running time.
- After controlling for differences in running time, action movies have a budget that is 2.25 times that of non-action movies, on average.

To better understand the nature of these effects, we find the fitted equations for action and non-action movies and then plot the two fitted curves using two `stat_function()` layers.

Non-Action

$$\begin{aligned}\widehat{\text{Budget}}_i &= e^{-5.56} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)} \times e^{0.81(0)} \\ &= e^{-5.56} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)}\end{aligned}$$

Action

$$\begin{aligned}\widehat{\text{Budget}}_i &= e^{-5.56} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)} \times e^{0.81(1)} \\ &= e^{-5.56} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)} \times e^{0.81} \\ &= e^{-5.56} \times e^{0.81} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)} \\ &= e^{-5.56+0.81} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)} \\ &= e^{-4.75} \times e^{0.12(\text{Running Time}_i)} \times e^{-0.0004(\text{Running Time}_i^2)}\end{aligned}$$

```
# Plot
ggplot(data = movies, aes(x = length, y = budget)) +
  geom_point(alpha = 0) +
  stat_function(fun = function(x) {exp(-5.56) * exp(0.12*x) * exp(-0.0004*x^2)},
    color = "black", linetype = "dashed") +
  stat_function(fun = function(x) {exp(-4.75) * exp(0.12*x) * exp(-0.0004*x^2)},
    color = "red", linetype = "solid") +
  theme_bw() +
  xlab("Running time (in minutes)") +
  ylab("Predicted budget (in millions of dollars)")
```

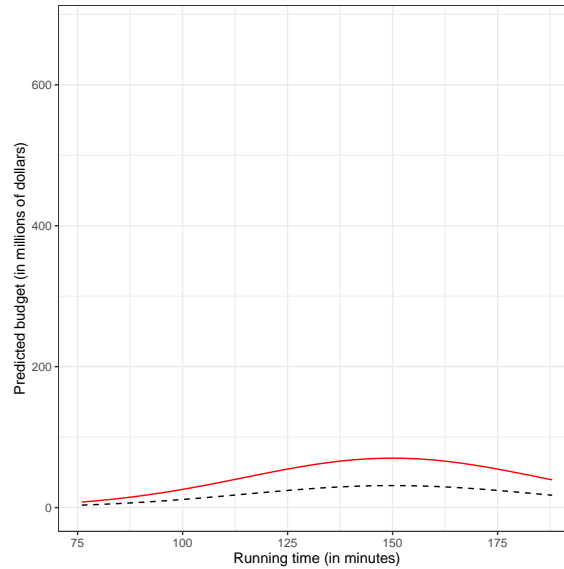


Figure 10. Plot of the predicted movie budget as a function of its running time for the polynomial model. The fitted curves for both action movies (red, solid line) and non-action movies (black, dashed line) are displayed.

This plot suggests that there is a curvilinear relationship between running time and log-budget for both action and non-action movies. In general, the effect of running time on log-budget increases for both types of movies with longer running times until about 150 minutes (approximate x -coordinate of the vertex) then the effect of running time has an increasing negative effect on log-budget. Furthermore, action movies have a higher average budget than non-action movies.

Note that although we fitted a main effects model, once we back-transform from an additive to a multiplicative relationship, the two fitted curves are no longer parallel. If we had plotted running time as a function of log-budget then the two curves would have been parallel.