Model Visualization

Dr Saufi

2024-05-29

Contents

1. Example Code Using the {jtools} Package

1.1 Loading Data and Fitting a Model

First, we need to load the necessary data and fit a regression model using the lm function.

```r
# Load the necessary package
library(jtools)

# Example dataset from jtools
data(movies)

# Fit a linear model
fit <- lm(metascore ~ imdb_rating + log(us_gross) + genre5, data = movies)
```

1.2 Summarizing the Model

We can summarize the model using the summ function to get a detailed output.

```r
# Summarize the model
summ(fit)
```

```
MODEL INFO:
Observations: 831 (10 missing obs. deleted)
Dependent Variable: metascore
Type: OLS linear regression

MODEL FIT:
F(6,824) = 169.37, p = 0.00
R² = 0.55
Adj. R² = 0.55

Standard errors: OLS
-----------------------------------------------------------
                                Est.   S.E.   t val.      p
--------------------------- -------- ------ -------- ------
(Intercept)                   -39.96   5.92    -6.75   0.00
imdb_rating                    12.80   0.49    25.89   0.00
log(us_gross)                   0.47   0.31     1.52   0.13
genre5Comedy                    6.32   1.06     5.95   0.00
genre5Drama                     7.66   1.08     7.12   0.00
genre5Horror/Thriller          -0.73   1.51    -0.48   0.63
genre5Other                     5.86   3.25     1.80   0.07
-----------------------------------------------------------
```

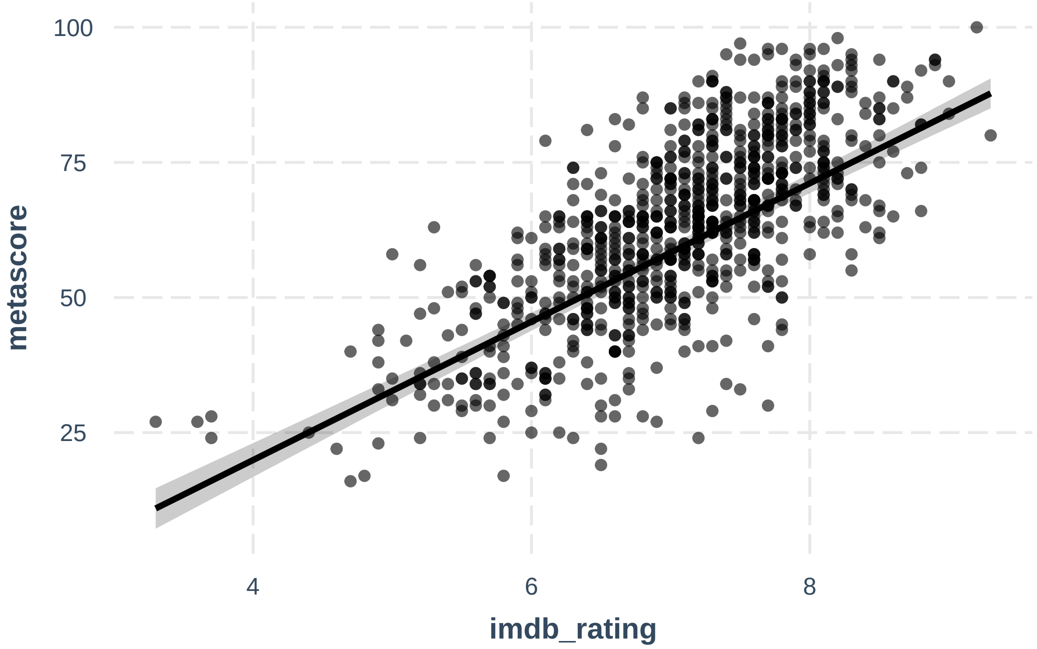
1.3 Visualizing Model Effects

The effect_plot function allows us to visualize the effect of a predictor variable on the response variable.

```r
# Visualize the effect of 'imdb_rating' on 'metascore'
effect_plot(fit, pred = imdb_rating, interval = TRUE, plot.points = TRUE)
```

```
Using data movies from global environment. This could cause incorrect
results if movies has been altered since the model was fit. You can
manually provide the data to the "data =" argument.
```

```
Warning: Removed 10 rows containing missing values or values outside the scale range
(`geom_point()`).
```



2. Example Code Using the {modelsummary} Package

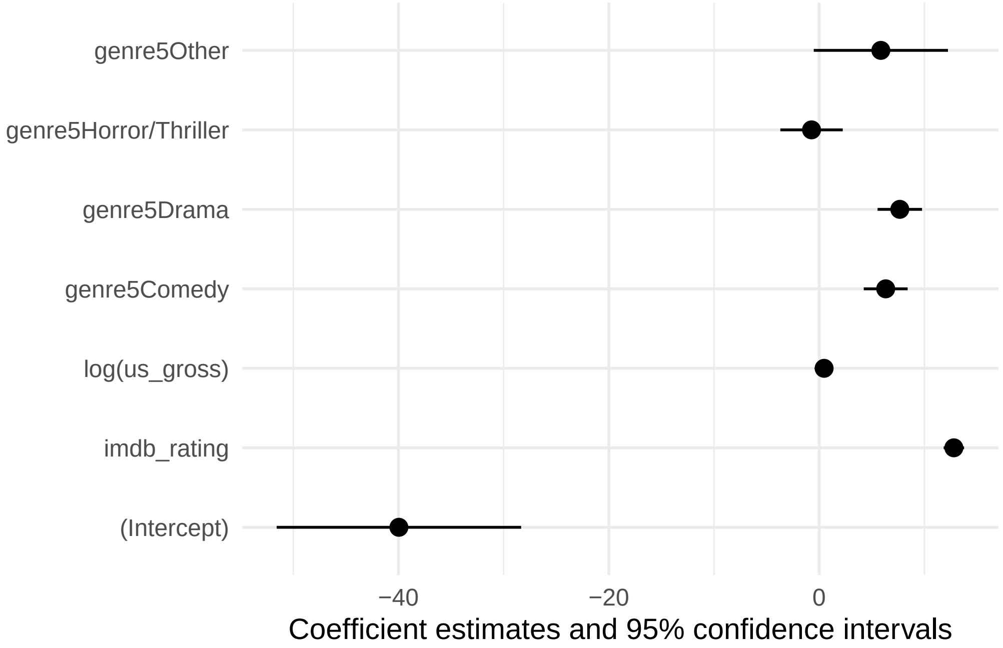
2.1 Visualizing a Single Model

The modelplot function from the modelsummary package can be used to create visualizations for a single model.

```
# Load the necessary package
library(modelsummary)
```

```
`modelsummary` 2.0.0 now uses `tinytable` as its default table-drawing
  backend. Learn more at: https://vincentarelbundock.github.io/tinytable/

Revert to `kableExtra` for one session:

  options(modelsummary_factory_default = 'kableExtra')
  options(modelsummary_factory_latex = 'kableExtra')
  options(modelsummary_factory_html = 'kableExtra')

Silence this message forever:

  config_modelsummary(startup_message = FALSE)
```

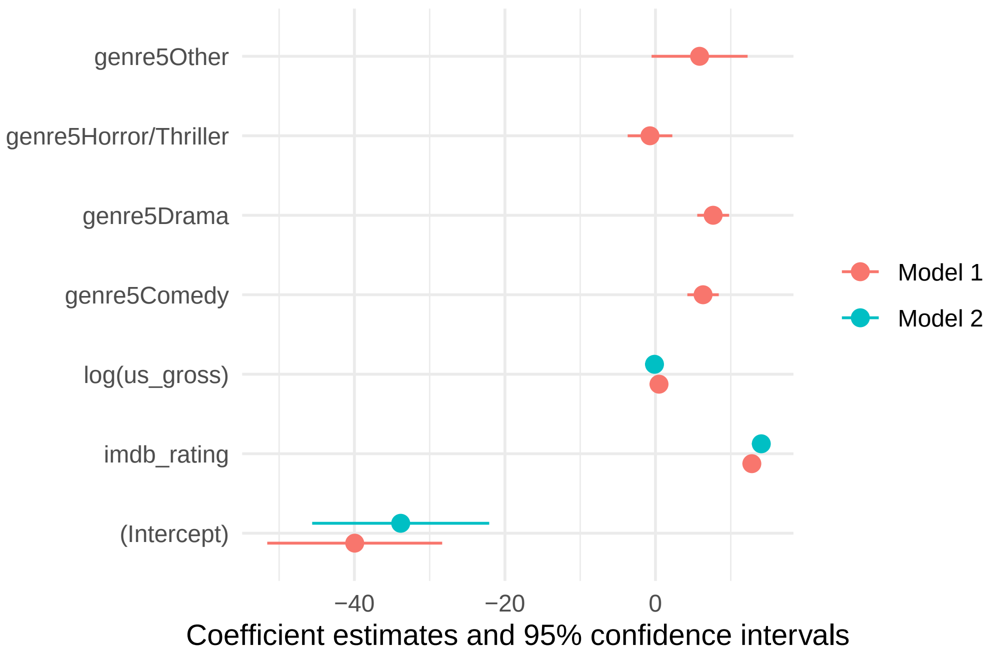
```
# Create a plot for the model
modelplot(fit)
```



2.2 Visualizing Multiple Models

If you have multiple models, you can visualize them together for comparison.

```
# Fit another model for comparison
fit2 <- lm(metascore ~ imdb_rating + log(us_gross), data = movies)

# Create a list of models
models <- list("Model 1" = fit, "Model 2" = fit2)

# Visualize multiple models
modelplot(models)
```


genre5Other
genre5Horror/Thriller
genre5Drama
genre5Comedy
log(us_gross)
imdb_rating
(Intercept)
−40
−20
0
Coefficient estimates and 95% confidence intervals
Model 1
Model 2