SehaKotkuKaya

2025-11-30

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.1     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.2.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(ggplot2)
library(stringr)

options(scipen = 999)
```

```
IMDB_Top_250_Movies <- read.csv("IMDB Top 250 Movies.csv", check.names = FALSE)

plot_data <- IMDB_Top_250_Movies %>%
  mutate(budget_raw = as.numeric(str_replace_all(budget, "[^0-9]", "")),

    budget_numeric = case_when(

      name == "Princess Mononoke" ~ 23500000,
      name == "3 Idiots" ~ 12000000,
      name == "Akira" ~ 10000000,
      name == "Spirited Away" ~ 19000000,
      name == "Dangal" ~ 9700000,
      name == "Grave of the Fireflies" ~ 3700000,
      TRUE ~ budget_raw
    ),
```

```
    box_office_clean = as.numeric(str_replace_all(box_office, "[^0-9]", ""))
  ) %>%

  filter(!is.na(budget_numeric) & budget_numeric > 0)
```

```
library(dplyr)

plot_data$box_office_clean <- as.numeric(gsub("[\\$,]", "", plot_data$box_office))
```

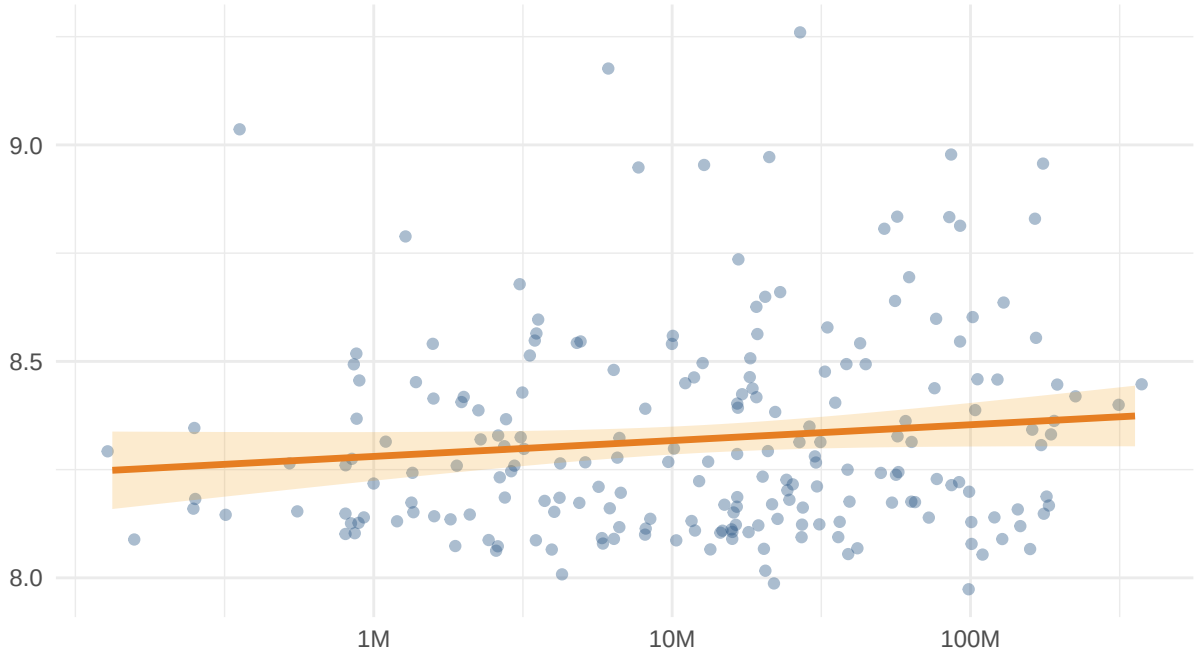
```
## Warning: NAs introduced by coercion
```

```
plot_data$budget_numeric <- as.numeric(gsub("[\\$,]", "", plot_data$budget_numeric))

plot_data <- plot_data %>%
  mutate(box_office_clean = case_when(
    name == "12 Angry Men" ~ 20000000,
     name == "Cool Hand Luke" ~ 670000,
    name == "Paths of Glory" ~ 525200,
    name == "Witness for the Prosecution" ~ 7000000,
    name == "Dersu Uzala" ~ 5000000,
    TRUE ~ box_office_clean




  ))
```

```
ggplot(plot_data, aes(x = budget_numeric, y = rating)) +
 geom_jitter(alpha = 0.4, color = "#2E5A88", width = 0.05, height = 0.05) +
  geom_smooth(method = "lm", color = "#E67E22", fill = "#F39C12", alpha = 0.2, size = 1.2) +
  scale_x_log10(
    breaks = c(1e6, 1e7, 1e8, 1e9),
    labels = c("1M", "10M", "100M", "1B")
  ) +

  labs(
    title = "Relationship Between Budget and IMDb Rating",
    subtitle = "Higher budgets don't always guarantee higher ratings",
    x = "",
    y = "",
    caption = "Note: Entries marked as 'Not Available' in the budget column (38 films)\nwere filtered ou
  ) +

  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Note: Entries marked as 'Not Available' in the budget column (38 films) were filtered out prior to analysis.

```
ggsave("butce_rating_analizi.png", width = 20, height = 15, units = "cm", dpi = 600) +
  theme(
    plot.title = element_text(face = "bold", size = 14, hjust = 0.5),
    plot.subtitle = element_text(size = 10, color = "gray30", hjust = 0.5),
    axis.title = element_text(face = "italic"),

  )
```

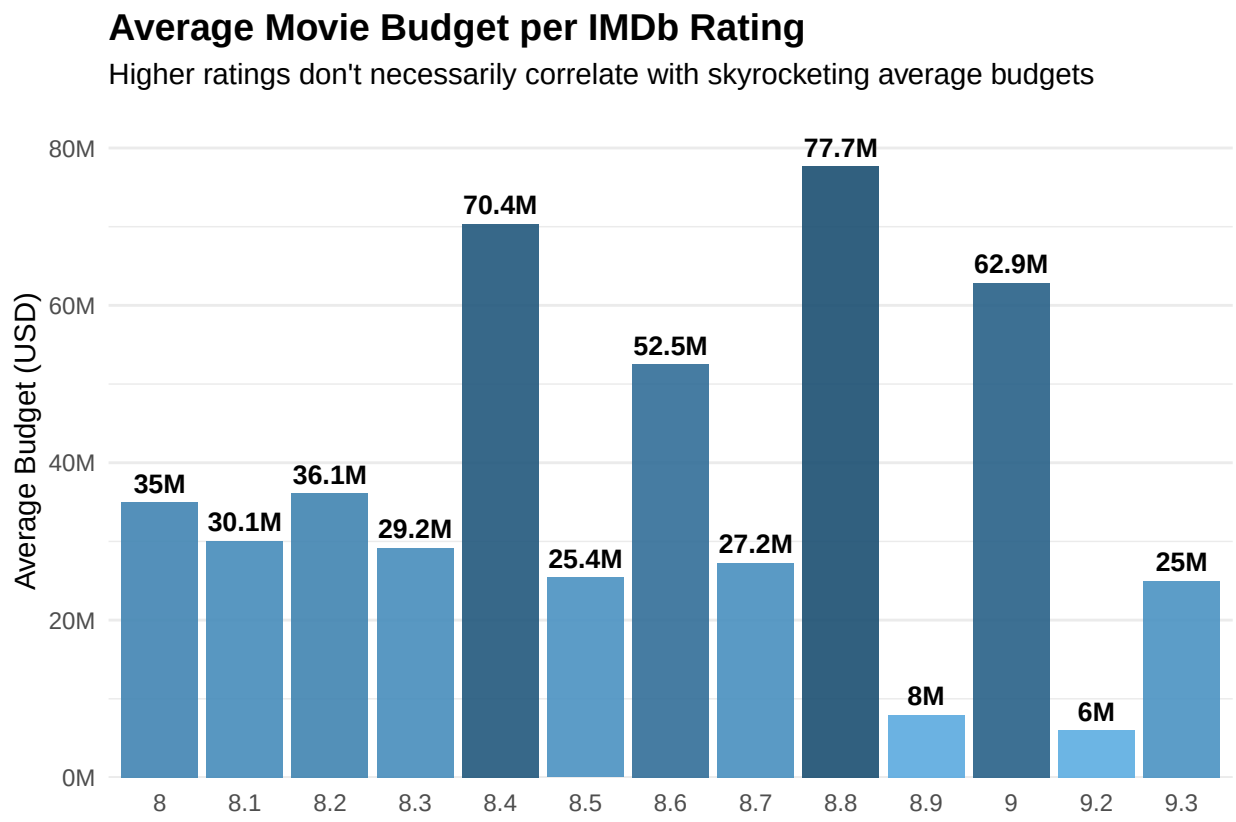
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## NULL
```

```
p2 <- ggplot(plot_data, aes(x = factor(rating), y = budget_numeric, fill = ..y..)) +
  stat_summary(fun = "mean", geom = "bar", alpha = 0.9) +
  stat_summary(fun = "mean", geom = "text", aes(label = paste0(round(..y.. / 1e6, 1), "M")),
               vjust = -0.5, size = 3.5, fontface = "bold") +
  scale_fill_gradient(low = "#5DADE2", high = "#1B4F72") +
  scale_y_continuous(
    labels = function(x) paste0(x / 1e6, "M"),
    expand = expansion(mult = c(0, 0.1))
  ) +
  labs(
    title = "Average Movie Budget per IMDb Rating",
    subtitle = "Higher ratings don't necessarily correlate with skyrocketing average budgets",
    x = "",
    y = "Average Budget (USD)",
    fill = "Avg Budget",
  ) +
```

```
  theme_minimal() +
  theme(
    legend.position = "none",
    panel.grid.major.x = element_blank(),
    plot.title = element_text(face = "bold", size = 14),
    axis.text.x = element_text(angle = 0, vjust = 0.5)
  )
print(p2)
```

```
## Warning: The dot-dot notation (`..y..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(y)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

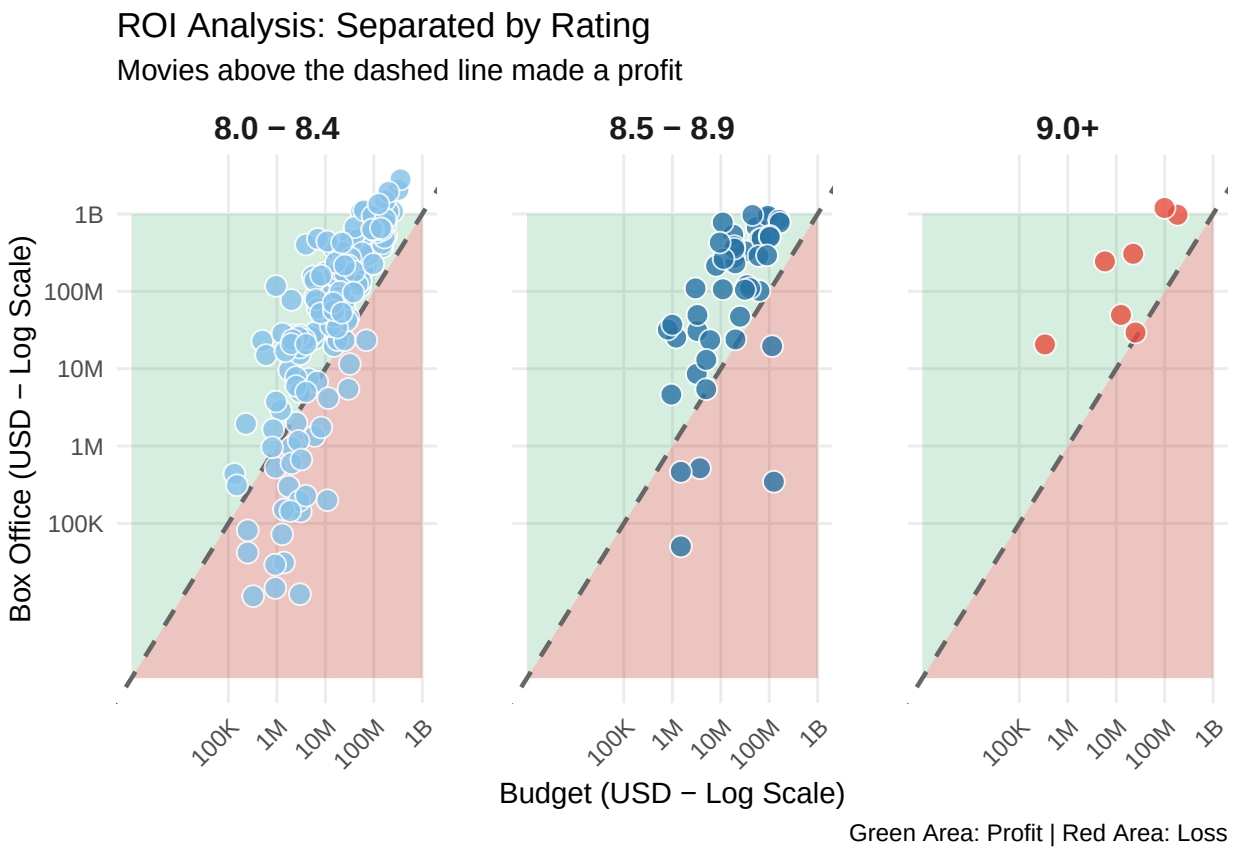



```
ggsave("ortbutce.png", plot = p2, width = 25, height = 18, units = "cm", dpi = 600)
```

```
plot_data <- plot_data %>%
  mutate(rating_group = case_when(
    rating >= 9.0 ~ "9.0+",
    rating >= 8.5 ~ "8.5 - 8.9",
    rating >= 8.0 ~ "8.0 - 8.4",
    TRUE ~ "8.0 Altı"
  ))

roi_data <- plot_data %>%
  filter(!is.na(budget_numeric), !is.na(box_office_clean),
         budget_numeric > 0, box_office_clean > 0)
```

```r
min_val <- 1000
max_val <- 1000000000

profit_poly <- data.frame(
  x = c(min_val, min_val, max_val),
  y = c(min_val, max_val, max_val)
)

loss_poly <- data.frame(
  x = c(min_val, max_val, max_val),
  y = c(min_val, min_val, max_val)
)
p_roi <- ggplot(roi_data, aes(x = budget_numeric, y = box_office_clean)) +
  geom_polygon(data = profit_poly, aes(x = x, y = y), fill = "#36AE60", alpha = 0.2) +
  geom_polygon(data = loss_poly, aes(x = x, y = y), fill = "#C0392B", alpha = 0.3) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "grey40", size = 0.8) +
  geom_jitter(aes(fill = rating_group), color = "white", shape = 21,
              size = 3.5, alpha = 0.8, stroke = 0.5) +

  facet_wrap(~rating_group) +
  scale_y_continuous(
    trans = "log10",
    breaks = c(1e5, 1e6, 1e7, 1e8, 1e9),
    labels = c("100K", "1M", "10M", "100M", "1B")
  ) +
  scale_x_continuous(
    trans = "log10",
    breaks = c(1e5, 1e6, 1e7, 1e8, 1e9),
    labels = c("100K", "1M", "10M", "100M", "1B")
  ) +

  # Renkler
  scale_fill_manual(values = c("8.0 - 8.4" = "#85C1E9",
                               "8.5 - 8.9" = "#2874A6",
                               "9.0+" = "#E74C3C",
                               "8.0 Altı" = "grey")) +

  labs(
    title = "ROI Analysis: Separated by Rating",
    subtitle = "Movies above the dashed line made a profit",
    x = "Budget (USD - Log Scale)",
    y = "Box Office (USD - Log Scale)",
    caption = "Green Area: Profit | Red Area: Loss"
  ) +

  theme_minimal() +
  theme(
    legend.position = "none",
    strip.text = element_text(size = 12, face = "bold"),
    panel.spacing = unit(1, "cm"),
    axis.text.x = element_text(angle = 45, hjust = 1),
    panel.grid.minor = element_blank()
  )
```

```
print(p_roi)
```




```
ggsave("roianly.png",
       width = 12,
       height = 8,
       dpi = 600,
       units = "in")
```

```
library(viridis)
```

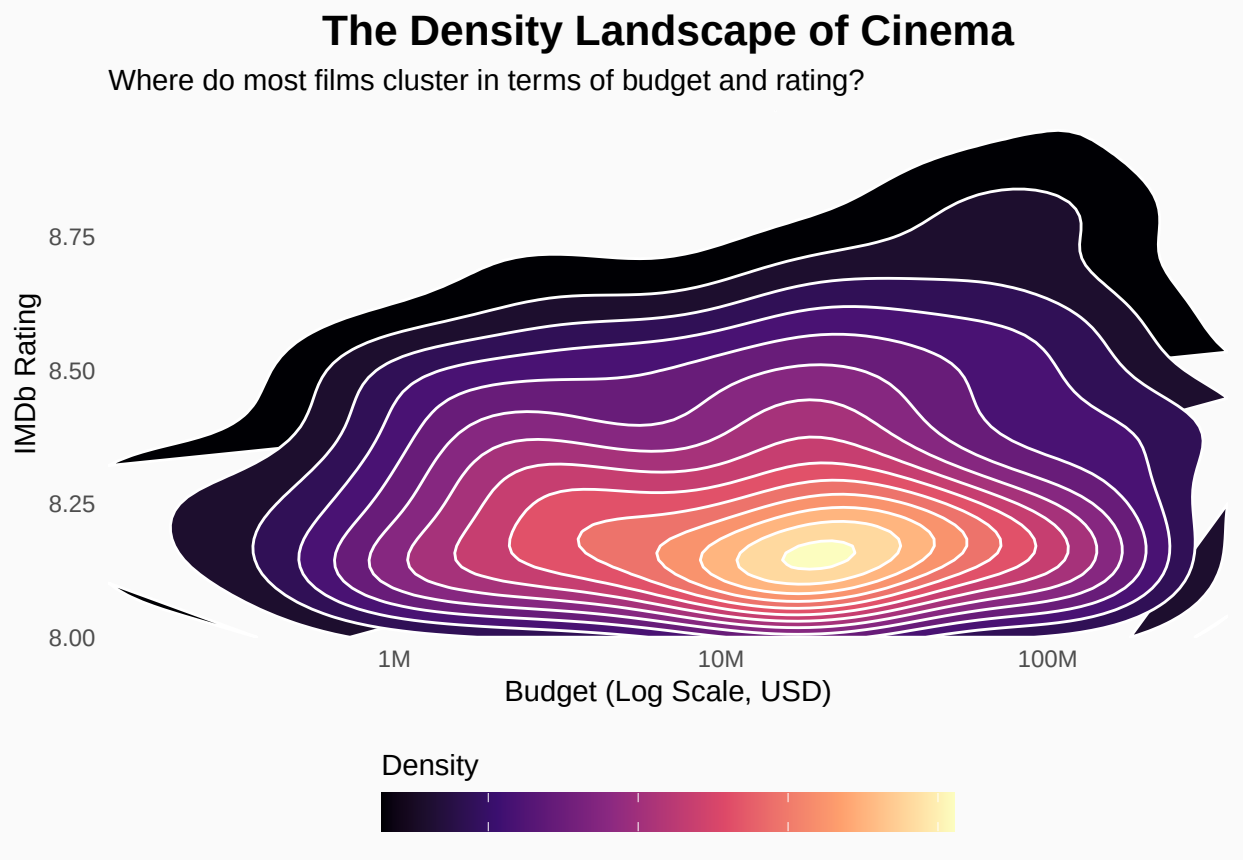
```
## Loading required package: viridisLite
```

```
##
## Attaching package: 'viridis'
```

```
## The following object is masked from 'package:scales':
##
##     viridis_pal
```

```
p4 <- ggplot(plot_data, aes(x = budget_numeric, y = rating)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", color = "white", bins = 15) +

  scale_x_log10(
    breaks = c(1e6, 1e7, 1e8, 1e9),
    labels = c("1M", "10M", "100M", "1B"),
    expand = c(0, 0)
  ) +
  scale_y_continuous(expand = c(0, 0)) +
```

```
  scale_fill_viridis_c(
    option = "magma",
    name = "Density",
    guide = guide_colorbar(
      direction = "horizontal",
      title.position = "top",
      barwidth = 15,
      barheight = 1,
      label = FALSE,
      ticks = FALSE
    )
  ) +

  labs(
    title = "The Density Landscape of Cinema",
    subtitle = "Where do most films cluster in terms of budget and rating?",
    x = "Budget (Log Scale, USD)",
    y = "IMDb Rating"
  ) +

  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 16, hjust = 0.5),
    panel.grid = element_blank(),
    legend.position = "bottom",
    plot.background = element_rect(fill = "grey98", color = NA)
  )
p4
```




```
ggsave("isi.png",
       width = 12,
       height = 8,
       dpi = 600,
       units = "in")
```