dashboard

```
library(readr)
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
library(ggplot2)
library(tidyr)
library(purrr)
library(jsonlite)
```

```
Attaching package: 'jsonlite'

The following object is masked from 'package:purrr':

    flatten
```

```r
df <- read_csv("data/tmdb_5000_movies.csv", show_col_types = FALSE)
df <- df |>
  mutate(
    release_date_parsed = suppressWarnings(ymd(release_date)),
    release_decade = case_when(
      year(release_date_parsed) >= 1980 & year(release_date_parsed) < 1990 ~ "1980s",
      year(release_date_parsed) >= 1990 & year(release_date_parsed) < 2000 ~ "1990s",
      year(release_date_parsed) >= 2000 & year(release_date_parsed) < 2010 ~ "2000s",
      year(release_date_parsed) >= 2010 & year(release_date_parsed) < 2020 ~ "2010s",
      year(release_date_parsed) >= 2020 & year(release_date_parsed) < 2030 ~ "2020s",
      TRUE ~ NA_character_
      ),
     english_group = ifelse(
      !is.na(original_language) & tolower(original_language) == "en",
      "English", "Non-English"
      )
  )

df_filter <- df |>
  filter(
    !is.na(release_date_parsed),
    !is.na(genres), genres != "", genres != "[]",
    !is.na(revenue), revenue >= 0,
    !is.na(budget),  budget  >= 0,
    !is.na(vote_average), vote_average >= 0, vote_average <= 10
  )
```

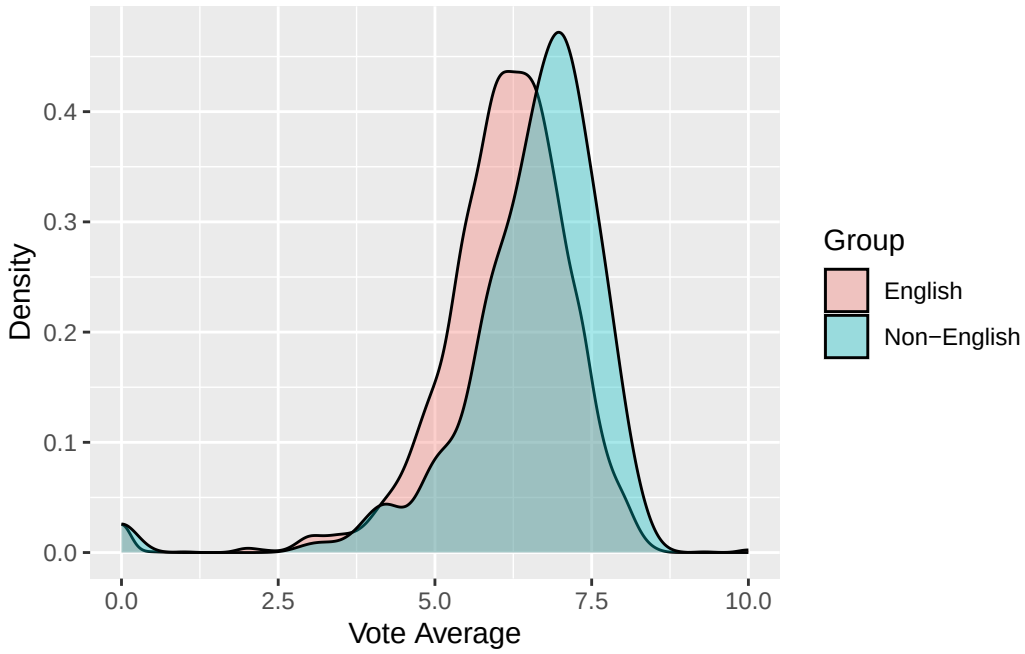
```r
b_ttest <- t.test(
  vote_average ~ english_group,
  data = df_filter,
  conf.level = 0.90
)
b_ttest
```

```
    Welch Two Sample t-test

data:  vote_average by english_group
t = -5.3673, df = 329.18, p-value = 1.512e-07
alternative hypothesis: true difference in means between group English and group Non-
English is not equal to 0
90 percent confidence interval:
 -0.5276455 -0.2795684
sample estimates:
    mean in group English mean in group Non-English
                 6.089010                  6.492617
```

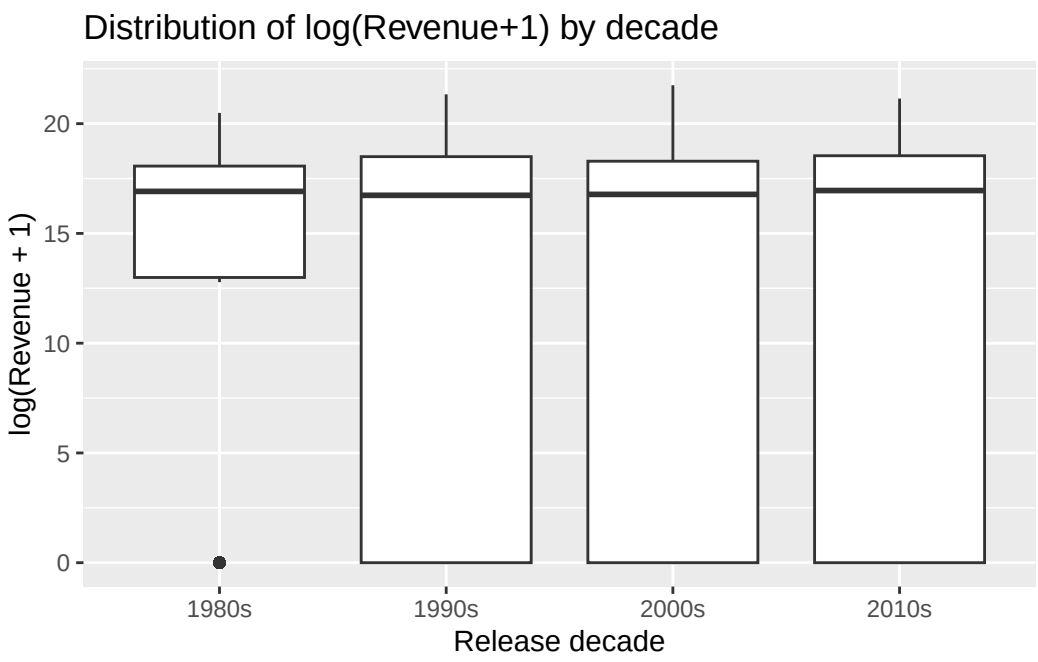
```r
df_filter |>
  group_by(english_group) |>
  summarize(
    n    = n(),
    mean = mean(vote_average),
    sd   = sd(vote_average),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 4
  english_group     n  mean    sd
  <chr>         <int> <dbl> <dbl>
1 English        4477  6.09  1.13
2 Non-English     298  6.49  1.27
```

```r
df_filter |>
  ggplot(aes(vote_average, fill = english_group)) +
  geom_density(alpha = 0.35) +
  labs(x = "Vote Average", y = "Density", fill = "Group")
```




```
df_c <- df_filter |>
  mutate(log_revenue = log1p(revenue)) |>
  filter(!is.na(release_decade)) |>
  group_by(release_decade) |>
  mutate(n_decade = n()) |>
  ungroup()

df_c |>
  ggplot(aes(x = release_decade, y = log_revenue)) +
  geom_boxplot() +
  labs(x = "Release decade", y = "log(Revenue + 1)",
       title = "Distribution of log(Revenue+1) by decade")
```




```
df_c |>
  count(release_decade, sort = TRUE)
```

```
# A tibble: 4 x 2
  release_decade     n
  <chr>          <int>
1 2000s           2041
2 2010s           1429
3 1990s            776
4 1980s            277
```

```
fit_c <- aov(log_revenue ~ release_decade, data = df_c)
summary(fit_c)
```

```
                Df Sum Sq Mean Sq F value Pr(>F)
release_decade    3    520   173.4   2.585 0.0515 .
Residuals      4519 303217    67.1
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(fit_c, conf.level = 0.90)
```

```
  Tukey multiple comparisons of means
    90% family-wise confidence level

Fit: aov(formula = log_revenue ~ release_decade, data = df_c)

$release_decade
                   diff        lwr         upr     p adj
1990s-1980s -0.7129820 -2.0270391  0.60107511 0.5990527
2000s-1980s -1.2745683 -2.4767390 -0.07239766 0.0716672
2010s-1980s -0.8541542 -2.0867035  0.37839513 0.3853463
2000s-1990s -0.5615863 -1.3533797  0.23020703 0.3641933
2010s-1990s -0.1411722 -0.9783701  0.69602567 0.9803849
2010s-2000s  0.4204141 -0.2271726  1.06800082 0.4448412
```

```r
tbl_c <- df_c |>
  group_by(release_decade) |>
  summarize(
    n    = n(),
    mean = mean(log_revenue),
    sd   = sd(log_revenue),
    .groups = "drop"
  )
tbl_c
```

```
# A tibble: 4 x 4
  release_decade     n  mean    sd
  <chr>          <int> <dbl> <dbl>
1 1980s            277  13.2  7.57
2 1990s            776  12.5  8.03
3 2000s           2041  11.9  8.30
4 2010s           1429  12.3  8.24
```

```r
df_d <- df_filter |>
  filter(!is.na(english_group)) |>
  mutate(profit = as.integer(revenue >2.4* budget)) |>
  select(english_group, profit)

tab_wide <- df_d |>
```

```
  mutate(english_group = factor(english_group, levels = c("English","Non-English"))) |>
  count(english_group, profit) |>
  pivot_wider(names_from = profit, values_from = n, values_fill = 0) |>
  rename(not_profit = `0`, profit = `1`) |>
  arrange(english_group)

tab_wide
```

```
# A tibble: 2 x 3
  english_group not_profit profit
  <fct>              <int>  <int>
1 English             2847   1630
2 Non-English          222     76
```

```
a <- tab_wide |> filter(english_group == "English")     |> pull(profit)
b <- tab_wide |> filter(english_group == "English")     |> pull(not_profit)
c <- tab_wide |> filter(english_group == "Non-English") |> pull(profit)
d <- tab_wide |> filter(english_group == "Non-English") |> pull(not_profit)


c(a=a, b=b, c=c, d=d)
```

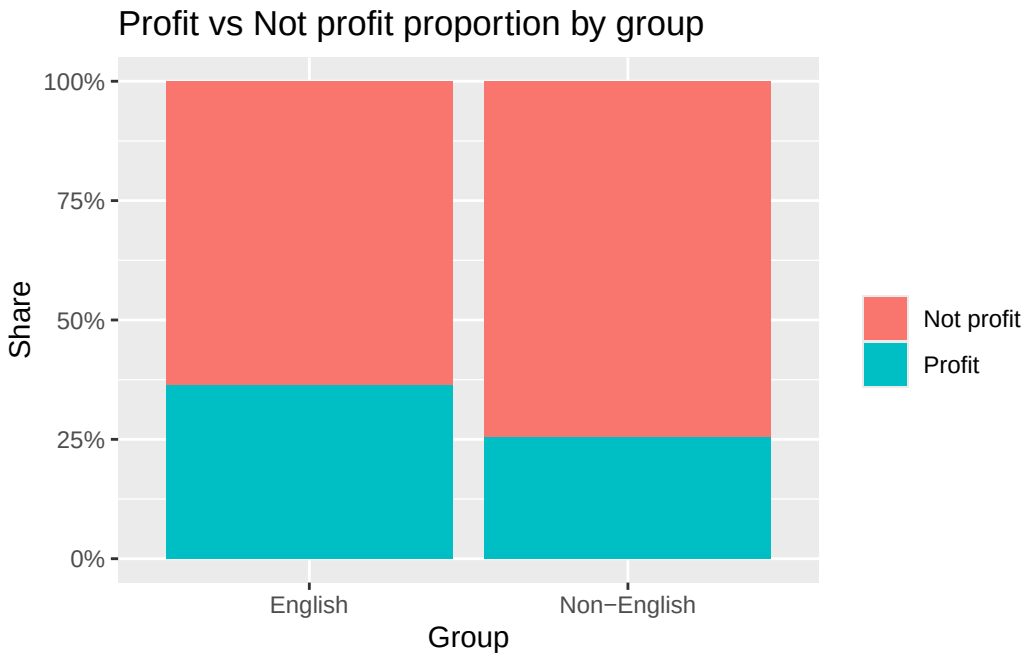
```
   a    b    c    d
1630 2847   76  222
```

```
p1 <- a/(a+b)
p2 <- c/(c+d)

rd_test <- prop.test(x = c(a,c), n = c(a+b, c+d), conf.level = 0.90, correct = TRUE)
rd_est  <- unname(p1 - p2)
rd_ci   <- unname(rd_test$conf.int)

RR <- p1/p2
se_logRR <- sqrt(1/a - 1/(a+b) + 1/c - 1/(c+d))
z <- qnorm(0.95)
RR_ci <- exp(log(RR) + c(-1,1)*z*se_logRR)

or_test <- fisher.test(matrix(c(a,b,c,d), nrow = 2), conf.level = 0.90)
OR <- unname(or_test$estimate)
OR_ci <- unname(or_test$conf.int)
```

```r
tibble::tibble(
  metric   = c("Risk_English (p1)", "Risk_NonEnglish (p2)", "RD = p1 - p2", "RR = p1/p2", "OR
  estimate = c(p1, p2, rd_est, RR, OR),
  ci90_lwr = c(NA, NA, rd_ci[1], RR_ci[1], OR_ci[1]),
  ci90_upr = c(NA, NA, rd_ci[2], RR_ci[2], OR_ci[2])
)
```

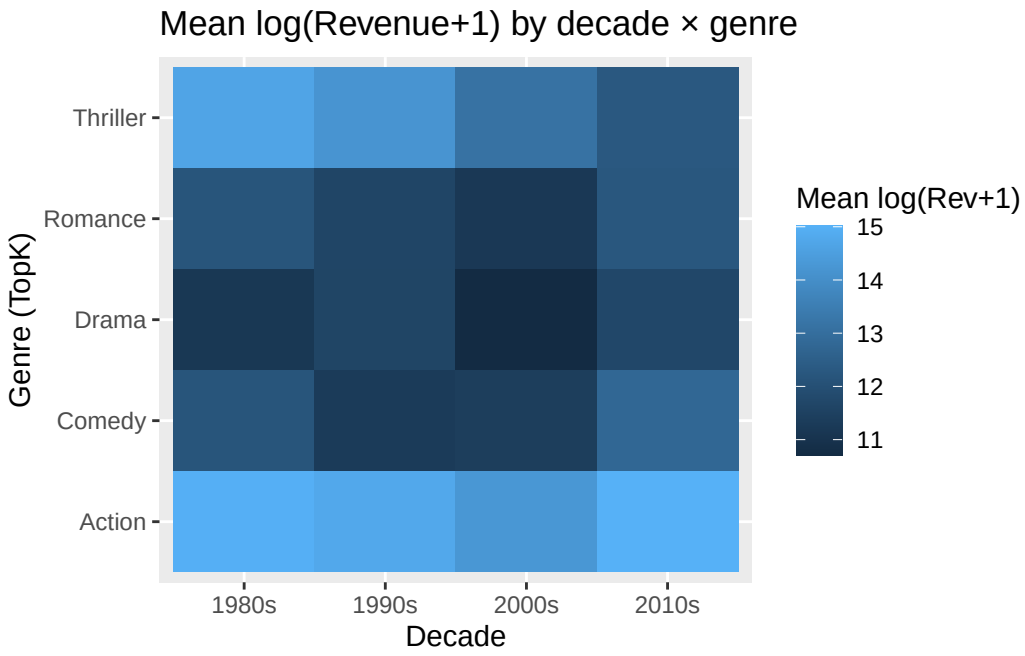
```
# A tibble: 5 x 4
  metric                estimate ci90_lwr ci90_upr
  <chr>                    <dbl>    <dbl>    <dbl>
1 Risk_English (p1)        0.364  NA        NA
2 Risk_NonEnglish (p2)     0.255  NA        NA
3 RD = p1 - p2             0.109   0.0641    0.154
4 RR = p1/p2               1.43    1.21      1.69
5 OR                       1.67    1.33      2.12
```

```r
df_d |>
  mutate(profit = factor(profit, levels = c(0,1), labels = c("Not profit","Profit"))) |>
  count(english_group, profit) |>
  group_by(english_group) |>
  mutate(pct = n / sum(n)) |>
  ungroup() |>
  ggplot(aes(x = english_group, y = pct, fill = profit)) +
  geom_col(position = "fill") +
  scale_y_continuous(labels = scales::percent) +
  labs(x = "Group", y = "Share", fill = "",
       title = "Profit vs Not profit proportion by group")
```




```r
top_k <- 5

safe_parse_genres <- function(s) {
  if (is.na(s) || s == "" || s == "[]") return(list())
  out <- tryCatch(jsonlite::fromJSON(s), error = function(e) NULL)
  if (is.null(out)) return(list())
  if (is.data.frame(out) && "name" %in% names(out)) {
    as.character(out$name)
  } else if (is.list(out)) {
    unlist(lapply(out, function(x) tryCatch(as.character(x$name), error = function(e) NA_cha
  } else character()
}

df_gen_long <- df_filter |>
  filter(!is.na(release_decade)) |>
  mutate(genres_vec = purrr::map(genres, safe_parse_genres)) |>
  tidyr::unnest_longer(genres_vec, values_to = "genre") |>
  filter(!is.na(genre), genre != "")

top_genres <- df_gen_long |>
  count(genre, sort = TRUE) |>
  slice_head(n = top_k) |>
  pull(genre)
```

```r
heat_rev_mean <- df_gen_long |>
  filter(genre %in% top_genres) |>
  mutate(
    release_decade = factor(release_decade, levels = c("1980s","1990s","2000s","2010s","2020s
    log_revenue = log1p(revenue)
  ) |>
  group_by(release_decade, genre) |>
  summarize(mean_log_rev = mean(log_revenue), .groups = "drop")

heat_rev_mean |>
  ggplot(aes(x = release_decade, y = genre, fill = mean_log_rev)) +
  geom_tile() +
  labs(x = "Decade", y = "Genre (TopK)",
       fill = "Mean log(Rev+1)",
       title = "Mean log(Revenue+1) by decade × genre")
```




```r
xtab_counts <- df_gen_long |>
  filter(genre %in% top_genres) |>
  count(release_decade, genre) |>
  tidyr::pivot_wider(names_from = genre, values_from = n, values_fill = 0) |>
  arrange(factor(release_decade, levels = c("1980s","1990s","2000s","2010s","2020s")))

xtab_counts
```

```
# A tibble: 4 x 6
  release_decade Action Comedy Drama Romance Thriller
  <chr>           <int>  <int> <int>   <int>    <int>
1 1980s              84     82   100      37       71
2 1990s             200    317   396     168      225
3 2000s             467    809  1015     436      536
4 2010s             345    461   646     195      396
```

```
xtab_rowprop <- xtab_counts |>
  tibble::column_to_rownames("release_decade") |>
  as.matrix() |>
  prop.table(margin = 1) |>
  as.data.frame() |>
  tibble::rownames_to_column("release_decade")

xtab_rowprop
```

```
  release_decade    Action    Comedy     Drama    Romance  Thriller
1          1980s 0.2245989 0.2192513 0.2673797 0.09893048 0.1898396
2          1990s 0.1531394 0.2427259 0.3032159 0.12863706 0.1722818
3          2000s 0.1431198 0.2479314 0.3110634 0.13361937 0.1642660
4          2010s 0.1688693 0.2256486 0.3162017 0.09544787 0.1938326
```

```
mat_counts <- xtab_counts |>
  tibble::column_to_rownames("release_decade") |>
  as.matrix()

chisq_res <- chisq.test(mat_counts)
```