SAP projekt

36552071 Kristijan Šagovac

2024-12-15

Učitajmo dataset i promotrimo što se u njemu nalazi:

```
filmovi <- read_csv("movie_IMDB.csv")
```

```
## Rows: 5043 Columns: 28
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (12): color, director_name, actor_2_name, genres, actor_1_name, movie_ti...
## dbl (16): num_critic_for_reviews, duration, director_facebook_likes, actor_3...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(filmovi)
```

```
## Rows: 5,043
## Columns: 28
## $ color                     <chr> "Color", "Color", "Color", "Color", NA, "Col~
## $ director_name             <chr> "James Cameron", "Gore Verbinski", "Sam Mend~
## $ num_critic_for_reviews    <dbl> 723, 302, 602, 813, NA, 462, 392, 324, 635, ~
## $ duration                  <dbl> 178, 169, 148, 164, NA, 132, 156, 100, 141, ~
## $ director_facebook_likes   <dbl> 0, 563, 0, 22000, 131, 475, 0, 15, 0, 282, 0~
## $ actor_3_facebook_likes    <dbl> 855, 1000, 161, 23000, NA, 530, 4000, 284, 1~
## $ actor_2_name              <chr> "Joel David Moore", "Orlando Bloom", "Rory K~
## $ actor_1_facebook_likes    <dbl> 1000, 40000, 11000, 27000, 131, 640, 24000, ~
## $ gross                     <dbl> 760505847, 309404152, 200074175, 448130642, ~
## $ genres                    <chr> "Action|Adventure|Fantasy|Sci-Fi", "Action|A~
## $ actor_1_name              <chr> "CCH Pounder", "Johnny Depp", "Christoph Wal~
## $ movie_title               <chr> "Avatar ", "Pirates of the Caribbean: At Wor~
## $ num_voted_users           <dbl> 886204, 471220, 275868, 1144337, 8, 212204, ~
## $ cast_total_facebook_likes <dbl> 4834, 48350, 11700, 106759, 143, 1873, 46055~
## $ actor_3_name              <chr> "Wes Studi", "Jack Davenport", "Stephanie Si~
## $ facenumber_in_poster      <dbl> 0, 0, 1, 0, 0, 1, 0, 1, 4, 3, 0, 0, 1, 2, 1,~
## $ plot_keywords             <chr> "avatar|future|marine|native|paraplegic", "g~
## $ movie_imdb_link           <chr> "http://www.imdb.com/title/tt0499549/?ref_=f~
## $ num_user_for_reviews      <dbl> 3054, 1238, 994, 2701, NA, 738, 1902, 387, 1~
## $ language                  <chr> "English", "English", "English", "English", ~
## $ country                   <chr> "USA", "USA", "UK", "USA", NA, "USA", "USA",~
## $ content_rating            <chr> "PG-13", "PG-13", "PG-13", "PG-13", NA, "PG-~
## $ budget                    <dbl> 237000000, 300000000, 245000000, 250000000, ~
```

```
## $ title_year                <dbl> 2009, 2007, 2015, 2012, NA, 2012, 2007, 2010~
## $ actor_2_facebook_likes    <dbl> 936, 5000, 393, 23000, 12, 632, 11000, 553, ~
## $ imdb_score                <dbl> 7.9, 7.1, 6.8, 8.5, 7.1, 6.6, 6.2, 7.8, 7.5,~
## $ aspect_ratio              <dbl> 1.78, 2.35, 2.35, 2.35, NA, 2.35, 2.35, 1.85~
## $ movie_facebook_likes      <dbl> 33000, 0, 85000, 164000, 0, 24000, 0, 29000,~
```

```
head(filmovi)
```

```
## # A tibble: 6 x 28
##   color director_name     num_critic_for_reviews duration director_facebook_li~1
##   <chr> <chr>                              <dbl>    <dbl>                  <dbl>
## 1 Color James Cameron                          723      178                      0
## 2 Color Gore Verbinski                         302      169                    563
## 3 Color Sam Mendes                             602      148                      0
## 4 Color Christopher Nolan                      813      164                  22000
## 5 <NA>  Doug Walker                             NA       NA                    131
## 6 Color Andrew Stanton                         462      132                    475
## # i abbreviated name: 1: director_facebook_likes
## # i 23 more variables: actor_3_facebook_likes <dbl>, actor_2_name <chr>,
## #   actor_1_facebook_likes <dbl>, gross <dbl>, genres <chr>,
## #   actor_1_name <chr>, movie_title <chr>, num_voted_users <dbl>,
## #   cast_total_facebook_likes <dbl>, actor_3_name <chr>,
## #   facenumber_in_poster <dbl>, plot_keywords <chr>, movie_imdb_link <chr>,
## #   num_user_for_reviews <dbl>, language <chr>, country <chr>, ...
```

Pitanje 4: Razlikuju li se ocjene s obzirom na vrijeme premijere?

Pošto nam za analizu ovog pitanja ne trebaju oni filmovi koji u stupcima "title_year" ili "imdb_score" imaju nedostajuću vrijednost, najprije ćemo njih izbaciti iz okvira. Također želimo izbaciti potencijalne duplikate.

```
filmovi4 <- filter(filmovi, !is.na(title_year) & !is.na(imdb_score))
filmovi4 <- unique(filmovi4)
```

Uzevši u obzir da u datasetu postoji 91 različita godina (te je shodno tome za neke od njih vrlo malen uzorak filmova), analizirat ćemo filmove po desetljećima u kojima su izašli. Dodat ćemo stupac "decade" u podatkovni okvir.

```
filmovi4 <- mutate(filmovi4, decade = title_year - title_year %% 10)
```

Provjerimo sada veličinu uzorka za svako desetljeće:

```
count(filmovi4, decade)
```

```
## # A tibble: 11 x 2
##    decade     n
##     <dbl> <int>
##  1   1910     1
##  2   1920     5
##  3   1930    15
##  4   1940    25
```

```
##  5   1950    28
##  6   1960    72
##  7   1970   112
##  8   1980   287
##  9   1990   782
## 10   2000  2083
## 11   2010  1481
```

Pošto u 1910-ima postoji samo jedan film, izbacit ćemo ga iz ove analize.

```
filmovi4 <- filter(filmovi4, decade != 1910)
```

Također, primjećujemo da su veličine uzoraka za neka desetljeća manja od 30. Kako bismo proveli ANOVA postupak, najprije moramo provjeriti jesu li ocjene po desetljećima normalno distribuirane. To možemo učiniti upotrebom Lillieforsovog testa, koji je inačica Kolmogorov-Smirnovljevog testa, ali se može upotrijebiti i ako varijanca i aritmetička sredina nisu poznate. Početna hipoteza testa je da je razdioba koju testiramo normalna, a alternativna je suprotna početnoj.

```
require(nortest)
```

```
## Loading required package: nortest
```

```
lillie.test(filmovi4$imdb_score[filmovi4$decade==1920])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1920]
## D = 0.31048, p-value = 0.1163
```

```
lillie.test(filmovi4$imdb_score[filmovi4$decade==1930])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1930]
## D = 0.24195, p-value = 0.0183
```

```
lillie.test(filmovi4$imdb_score[filmovi4$decade==1940])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1940]
## D = 0.085221, p-value = 0.9108
```

```
lillie.test(filmovi4$imdb_score[filmovi4$decade==1950])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1950]
## D = 0.17239, p-value = 0.03248
```

```r
lillie.test(filmovi4$imdb_score[filmovi4$decade==1960])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1960]
## D = 0.11605, p-value = 0.0176
```

```r
lillie.test(filmovi4$imdb_score[filmovi4$decade==1970])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1970]
## D = 0.090583, p-value = 0.02452
```

```r
lillie.test(filmovi4$imdb_score[filmovi4$decade==1980])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1980]
## D = 0.090417, p-value = 6.38e-06
```

```r
lillie.test(filmovi4$imdb_score[filmovi4$decade==1990])
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 1990]
## D = 0.060573, p-value = 3.682e-07
```

```r
lillie.test(filmovi4$imdb_score[filmovi4$decade==2000])
```

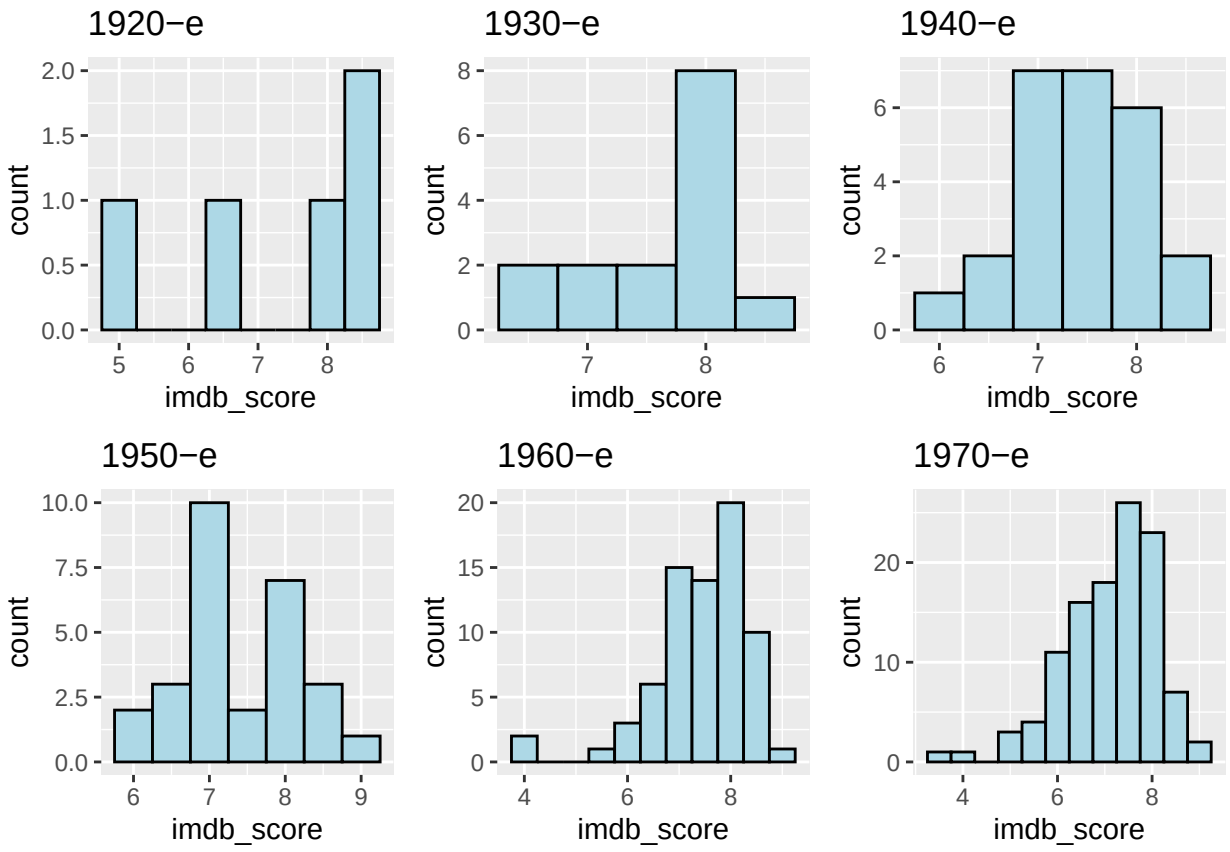
```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 2000]
## D = 0.071725, p-value < 2.2e-16
```

```r
lillie.test(filmovi4$imdb_score[filmovi4$decade==2010])
```

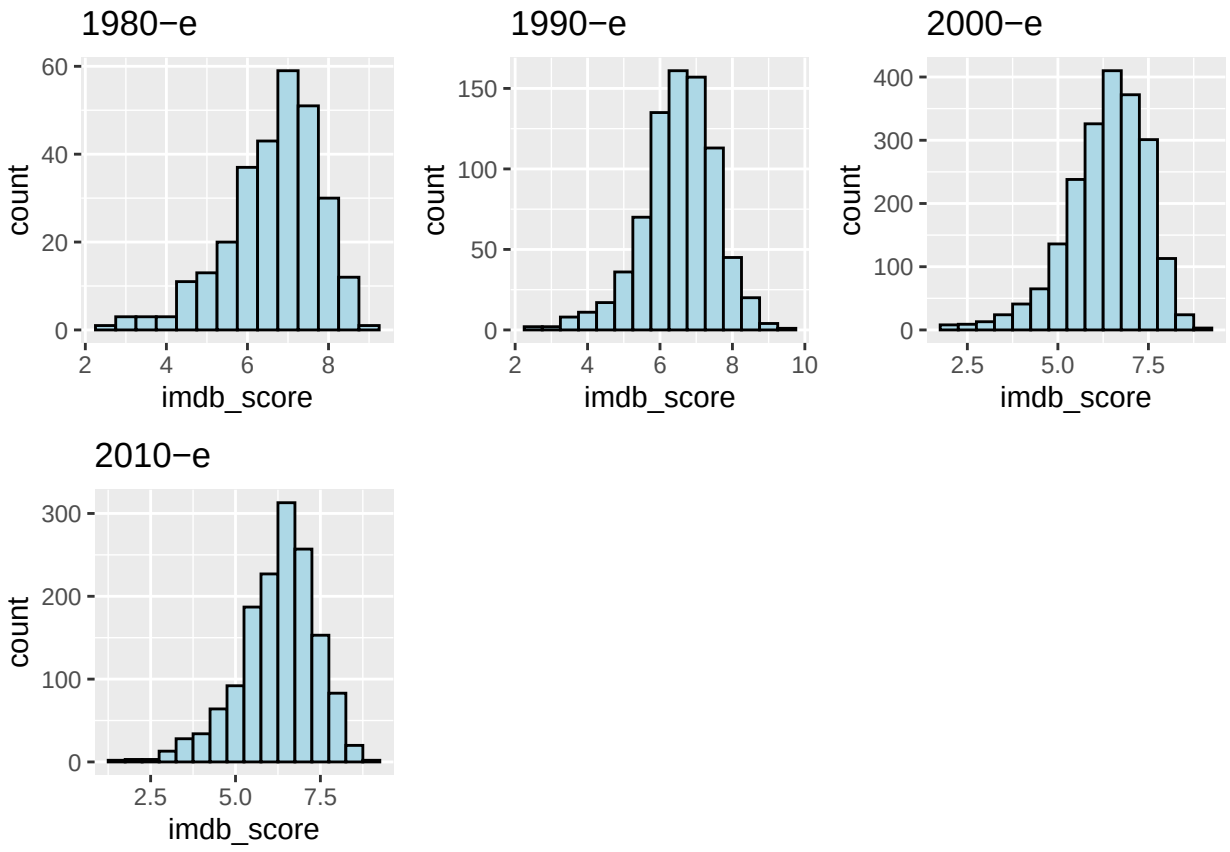
```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  filmovi4$imdb_score[filmovi4$decade == 2010]
## D = 0.077139, p-value < 2.2e-16
```

Uočimo da su p-vrijednosti za gotovo sva desetljeća iznimno male. Za sva desetljeća s p-vrijednostima manjima od 0.05 odbacujemo početnu hipotezu, odnosno zaključujemo da razdiobe ocjena filmovima po tim desetljećima nisu normalno distribuirane. Pogledajmo njihove razdiobe na sljedećim grafovima.

```r
graf0 <- ggplot(data = filmovi4[filmovi4$decade == 1920, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "1920-e")
graf1 <- ggplot(data = filmovi4[filmovi4$decade == 1930, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "1930-e")
graf2 <- ggplot(data = filmovi4[filmovi4$decade == 1940, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title ="1940-e")
graf3 <- ggplot(data = filmovi4[filmovi4$decade == 1950, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "1950-e")
graf4 <- ggplot(data = filmovi4[filmovi4$decade == 1960, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "1960-e")
graf5 <- ggplot(data = filmovi4[filmovi4$decade == 1970, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "1970-e")
graf6 <- ggplot(data = filmovi4[filmovi4$decade == 1980, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "1980-e")
graf7 <- ggplot(data = filmovi4[filmovi4$decade == 1990, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "1990-e")
graf8 <- ggplot(data = filmovi4[filmovi4$decade == 2000, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "2000-e")
graf9 <- ggplot(data = filmovi4[filmovi4$decade == 2010, ], aes(x = imdb_score)) +
    geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
    labs(title = "2010-e")
grid.arrange(graf0, graf1, graf2, graf3, graf4, graf5, ncol = 3)
```




```
grid.arrange(graf6, graf7, graf8, graf9, ncol = 3)
```

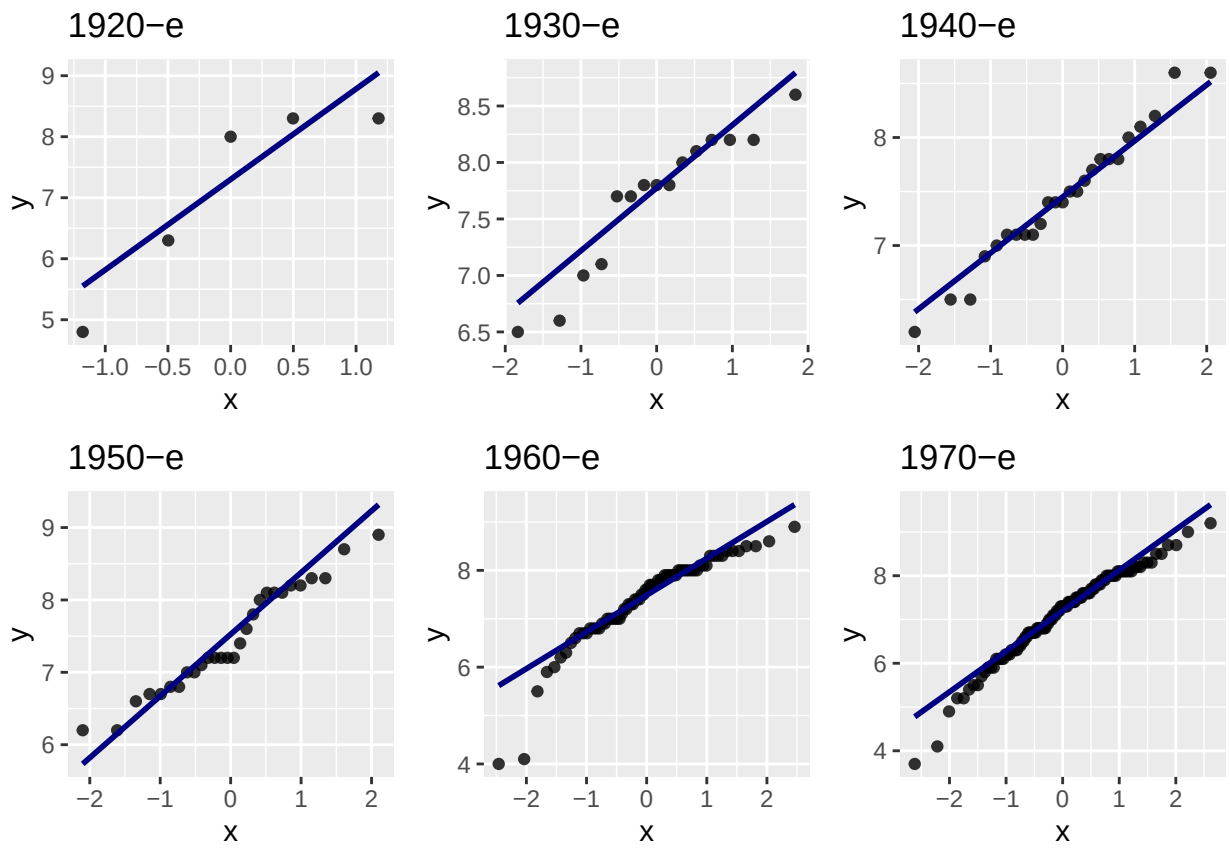


Histogrami pokazuju asimetriju u distribuciji. To ćemo još bolje prikazati na sljedećim QQ grafovima:

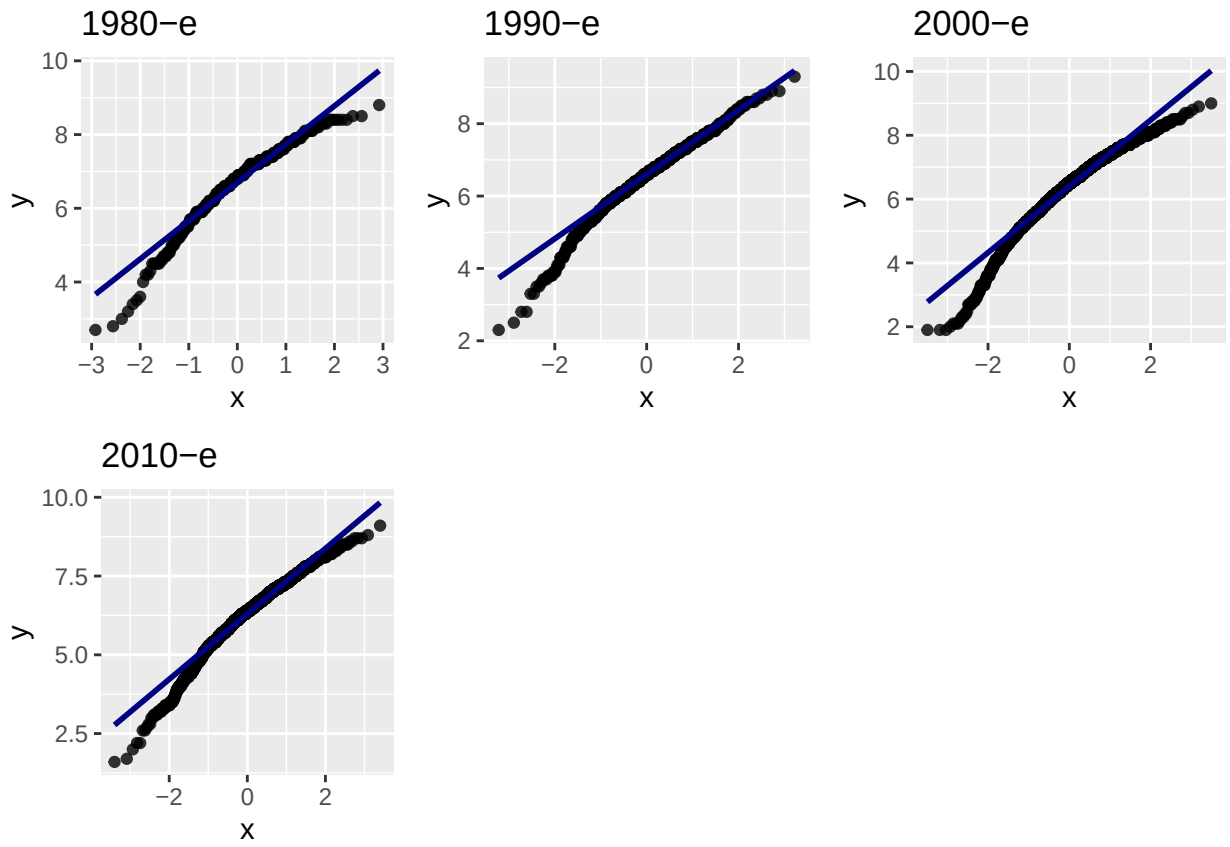
```
graf0 <- ggplot(filmovi4[filmovi4$decade==1920, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
```

```
## 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.
```

```
graf1 <- ggplot(filmovi4[filmovi4$decade==1930, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf2 <- ggplot(filmovi4[filmovi4$decade==1940, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf3 <- ggplot(filmovi4[filmovi4$decade==1950, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf4 <- ggplot(filmovi4[filmovi4$decade==1960, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf5 <- ggplot(filmovi4[filmovi4$decade==1970, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf6 <- ggplot(filmovi4[filmovi4$decade==1980, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf7 <- ggplot(filmovi4[filmovi4$decade==1990, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf8 <- ggplot(filmovi4[filmovi4$decade==2000, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s
graf9 <- ggplot(filmovi4[filmovi4$decade==2010, ], aes(sample = imdb_score)) + geom_qq(alpha = 0.8) + s

grid.arrange(graf0, graf1, graf2, graf3, graf4, graf5, ncol = 3)
```




```
grid.arrange(graf6, graf7, graf8, graf9, ncol = 3)
```



Kada bi distribucije bile normalne, podatci na QQ grafu bi bili posloženi približno u ravnoj liniji, dok na ovim grafovima primjećujemo zakrivljenost tih linija. Time smo i grafički prikazali kako distribucije ocjena filmova po desetljećima nisu normalne.

Učinimo još Bartlettov test za provjeru homogenosti varijanci. Početna hipoteza je da su varijance za sva desetljeća jednake, dok je alternativna hipoteza suprotna.

```
bartlett.test(filmovi4$imdb_score ~ filmovi4$decade)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  filmovi4$imdb_score by filmovi4$decade
## Bartlett's K-squared = 42.686, df = 9, p-value = 2.462e-06
```

Iz navedenih testova uočavamo kako razdiobe po većini desetljeća nisu ni približno normalne, a varijance nisu homogene. To znači da trebamo upotrijebiti neparametarsku alternativu ANOVA postupku, a to je Kruskal-Wallisov test. Početna hipoteza testa je da su aritmetičke sredine ocjena u svim desetljećima jednake. Alternativna hipoteza je da se barem jedna od tih sredina razlikuje.

```
kruskal.test(imdb_score ~ decade, data = filmovi4)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  imdb_score by decade
## Kruskal-Wallis chi-squared = 250.22, df = 9, p-value < 2.2e-16
```

Izuzetno mala p-vrijednost nas upućuje da odbacimo početnu hipotezu. Time zaključujemo da se IMDB ocjene filmova razlikuju s obzirom na vrijeme premijere filma. Promotrimo na koji način se razlikuju, odnosno na koji način ocjena ovisi o vremenu premijere filma.

Faktorizirajmo stupac za desetljeće:

```
decades <- paste(seq(1920, 2010, by=10), seq(1929, 2019, by=10), sep="-")
filmovi4$decade <- factor(filmovi4$decade, levels = seq(1920, 2010, 10),
                          labels = decades)
```

Napravimo linearni model za dane podatke:

```
model = lm(imdb_score ~ decade, data = filmovi4)
summary(model)
```

```
##
## Call:
## lm(formula = imdb_score ~ decade, data = filmovi4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6516 -0.6159  0.1449  0.7484  2.8484
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.14000    0.48672  14.670   <2e-16 ***
## decade1930-1939  0.54667    0.56201   0.973   0.3308
## decade1940-1949  0.30400    0.53317   0.570   0.5686
## decade1950-1959  0.31714    0.52839   0.600   0.5484
## decade1960-1969  0.26000    0.50333   0.517   0.6055
## decade1970-1979 -0.01946    0.49746  -0.039   0.9688
## decade1980-1989 -0.49505    0.49094  -1.008   0.3133
## decade1990-1999 -0.62414    0.48827  -1.278   0.2012
## decade2000-2009 -0.78489    0.48730  -1.611   0.1073
## decade2010-2019 -0.88841    0.48754  -1.822   0.0685 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.088 on 4880 degrees of freedom
## Multiple R-squared:  0.04613,    Adjusted R-squared:  0.04437
## F-statistic: 26.22 on 9 and 4880 DF,  p-value: < 2.2e-16
```

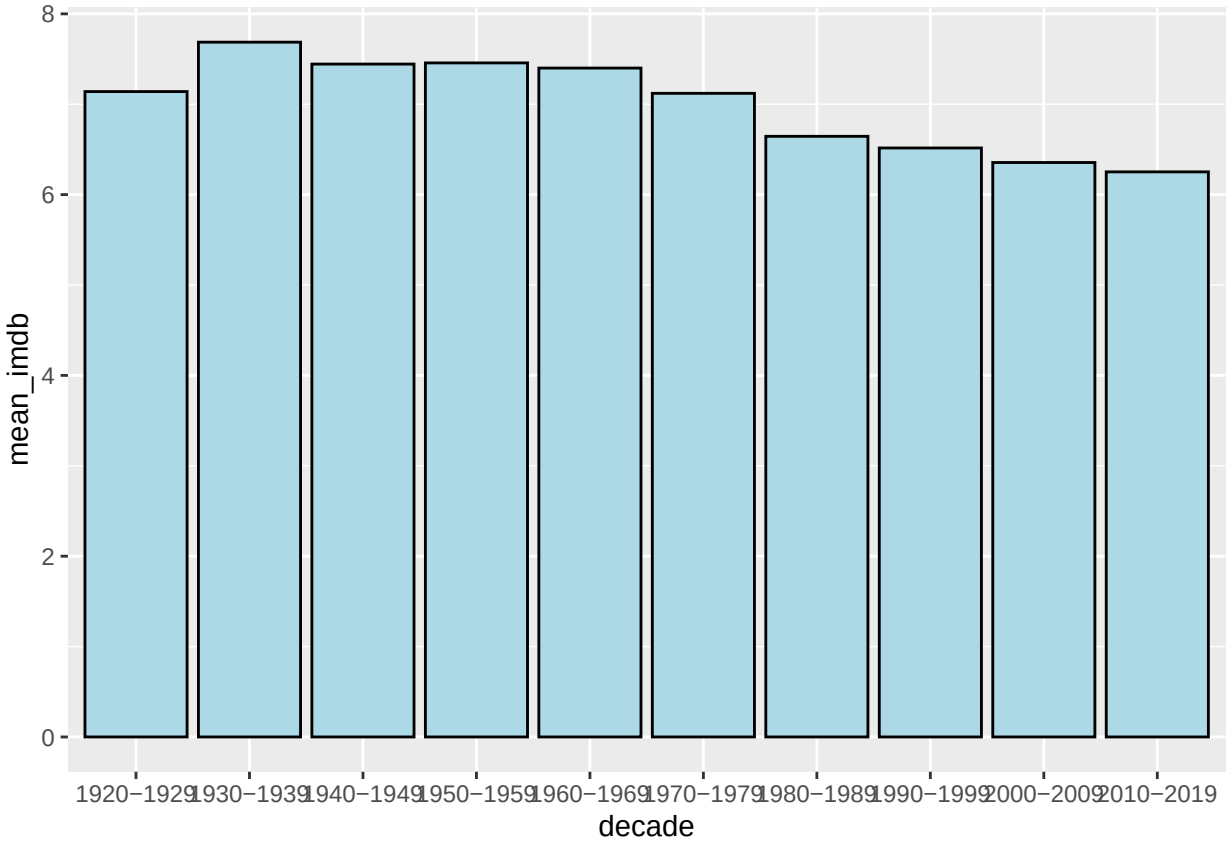
Prema procjeni ovog modela, prosječna IMDB ocjena filmova raste po desetljećima do 1930-ih, nakon kojih počinje padati. Pogledajmo kakav rezultat bismo dobili kad bismo prema njemu izvršili ANOVA postupak.

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: imdb_score
##             Df Sum Sq Mean Sq F value    Pr(>F)
## decade       9  279.5 31.0573   26.22 < 2.2e-16 ***
## Residuals 4880 5780.2  1.1845
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Iz navedenog postupka vidimo kako je p-vrijednost ponovno izuzetno malena, odnosno da se prosječna ocjena filmova razlikuje s obzirom na desetljeće. Rezultat testa slaže se s gore provedenim neparametarskim postupkom. Pogledajmo sada podatke po desetljećima na određenim grafovima kako bismo vizualno predočili tu razliku. Najprije ćemo prikazati histogram prosječnih ocjena po desetljećima.

```
f4_mean_scores <- filmovi4 %>%
  group_by(decade) %>%
  summarise(mean_imdb = mean(imdb_score, na.rm = TRUE))

ggplot(f4_mean_scores, aes(x = decade, y = mean_imdb)) + geom_bar(stat = "identity", fill = "lightblue"
```



Kao što je gornji model pokazao u svojim očekivanim vrijednostima, nakon 1930-ih primjećuje se pad prosječne ocjene. Prikažimo sada box-plot dijagram razdioba:

```
ggplot(filmovi4, aes(x = decade, y = imdb_score)) +
  geom_boxplot() +
  labs(x = "Desetljeće", y = "IMDb ocjena", title = "IMDb ocjene po desetljećima")
```

Primjetimo da osim što medijan načelno pada s obzirom na desetljeće, kasnija desetljeća imaju nekoliko stršećih vrijednosti sa jako niskim ocjenama, što nas upućuje da konačno razmislimo o pitanju: znači li ovo uistinu da su stariji filmovi i bolji? Pogledajmo kako se prosječne ocjene odnose prema broju filmova po desetljeću koji su uzeti u obzir. Već smo na početku ispisali te brojeve radi određivanja veličine uzoraka, ali radi jednostavnosti ćemo ih prikazati i ovdje.

```
film_count <- filmovi4 %>%
  group_by(decade) %>%
  summarise(
    count = n(),
    avg_score = mean(imdb_score)
  )

film_count
```

```
## # A tibble: 10 x 3
##    decade    count avg_score
##    <fct>     <int>     <dbl>
##  1 1920-1929     5      7.14
##  2 1930-1939    15      7.69
##  3 1940-1949    25      7.44
##  4 1950-1959    28      7.46
##  5 1960-1969    72      7.4
##  6 1970-1979   112      7.12
##  7 1980-1989   287      6.64
##  8 1990-1999   782      6.52
```

```
##  9 2000-2009  2083      6.36
## 10 2010-2019  1481      6.25
```

Dodajmo ih u tablicu filmova zajedno s prosjecima po desetljećima:

```
filmovi4 <- filmovi4 %>%
  left_join(film_count, by = "decade")
```

Napravimo model koji će pokušati predvidjeti prosječnu ocjenu filma po desetljeću s obzirom na broj filmova iz tog desetljeća.

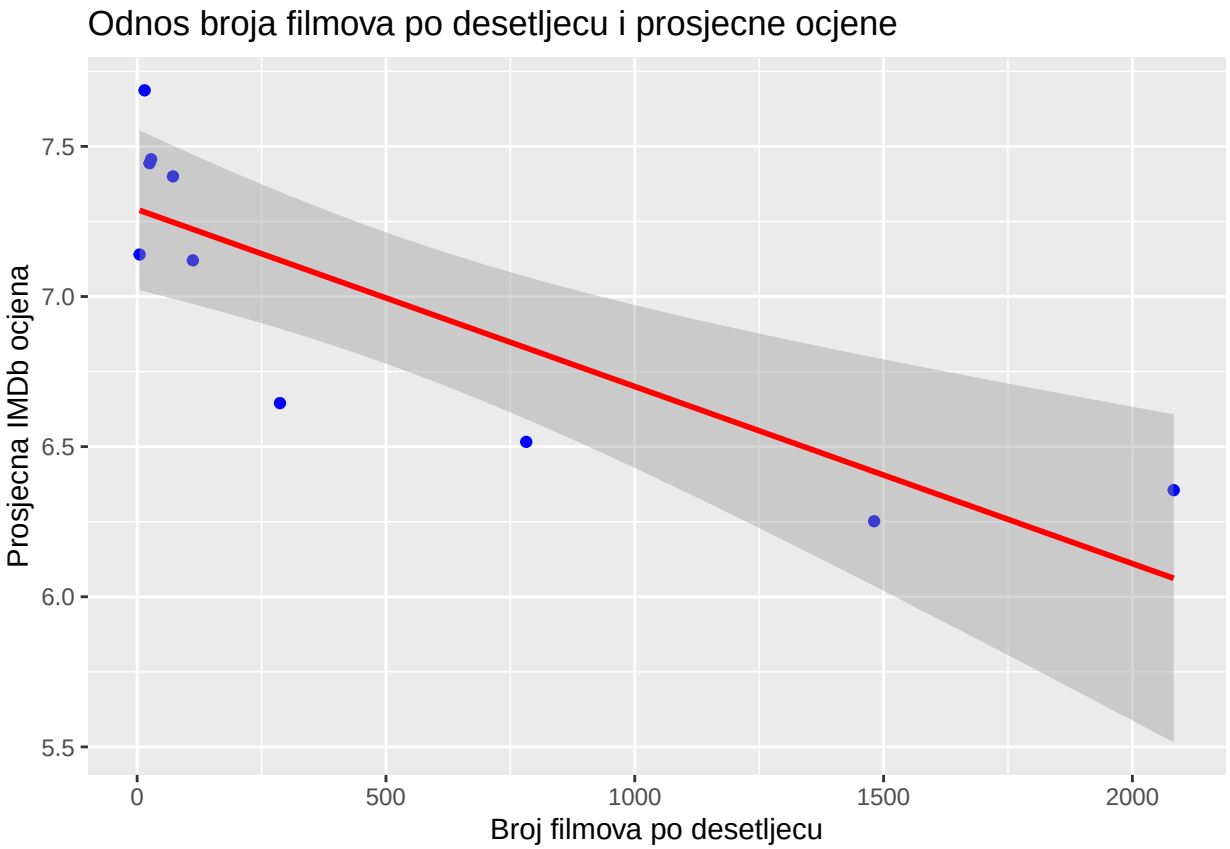
```
model_with_count <- lm(avg_score ~ count, data = filmovi4)

summary(model_with_count)
```

```
##
## Call:
## lm(formula = avg_score ~ count, data = filmovi4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16475 -0.16475 -0.07962  0.09418  0.89188
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.799e+00  6.095e-03  1115.4   <2e-16 ***
## count       -2.582e-04  3.769e-06   -68.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1708 on 4888 degrees of freedom
## Multiple R-squared:  0.4898, Adjusted R-squared:  0.4897
## F-statistic:  4692 on 1 and 4888 DF,  p-value: < 2.2e-16
```

Primjetimo da s obzirom na dane podatke, prosječna ocjena filmova iz pojedinog desetljeća uistinu ovisi o broju filmova uzetih u obzir iz tog desetljeća. Osim toga, model objašnjava gotovo polovicu varijance među prosječnim vrijednostima, iz čega možemo zaključiti da postoji trend smanjenja prosječnih ocjena po desetljeću što je više filmova u njemu. Prikažimo to i grafički.

```
ggplot(film_count, aes(x = count, y = avg_score)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(
    title = "Odnos broja filmova po desetljeću i prosječne ocjene",
    x = "Broj filmova po desetljeću",
    y = "Prosječna IMDb ocjena"
  )
```

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



Razlog ovakvog silaznog trenda može biti laka dostupnost današnjih filmova. Time je dostupno i više lošijih filmova, dok su od starih filmova dostupni samo oni koji su okarakterizirani kao klasici (dok se oni koji su bili smatrani lošijima u tadašnje vrijeme nisu sačuvali). Osim toga, moguće je da današnje filmove gleda šira publika raznovrsnijih filmskih ukusa, čime dolazi do veće varijabilnosti u korisničkim ocjenama.