Розрахункова робота №2

Завантажимо необхідні бібліотеки та дані.

```
library(ggplot2)
library(scales)
library(dplyr)
library(lubridate)
library(openair)
library(pastecs)
library(psych)
library(Mcomp)
library(xts)
library(zoo)
library(TTR)
library(timeDate)
library(tseries)
library(rlist)
library(knitr)
library(skimr)
library(readr)
library(magrittr)
library(tidyr)
library(corrplot)
library(circlize)
library(ggpubr)
library(forecast)
library(h2o)
```

```
movie_weekend <- read.delim("C:/Users/danie/Downloads/movieweekend-dat.txt", TRUE)
movie_daily <- read.delim("C:/Users/danie/Downloads/moviedaily-dat.txt", TRUE)
movie_total <- read.delim("C:/Users/danie/Downloads/movietotal-dat.txt", TRUE)
```

```r
head(movie_weekend, 7)
```

```
##   NUMBER            MOVIE WEEK_NUM WEEKEND_PER_THEATER WEEKEND_DATE
## 1      1 A Beautiful Mind        1                 701   12/21/2001
## 2      1 A Beautiful Mind        2               14820   12/28/2001
## 3      1 A Beautiful Mind        3                8940     1/4/2002
## 4      1 A Beautiful Mind        4                6850    1/11/2002
## 5      1 A Beautiful Mind        5                5280    1/18/2002
## 6      1 A Beautiful Mind        6                5155    1/25/2002
## 7      1 A Beautiful Mind        7                3735     2/1/2002
```

```r
head(movie_daily, 7)
```

```
##   NUMBER            MOVIE DAY_NUM DAILY_PER_THEATER       DATE
## 1      1 A Beautiful Mind       1              8909 12/24/2001
## 2      1 A Beautiful Mind       2              3885 12/25/2001
## 3      1 A Beautiful Mind       3              3365 12/26/2001
## 4      1 A Beautiful Mind       4              3324 12/27/2001
## 5      1 A Beautiful Mind       5              4403 12/28/2001
## 6      1 A Beautiful Mind       6              5475 12/29/2001
## 7      1 A Beautiful Mind       7              4964 12/30/2001
```

```r
head(movie_total, 7)
```

```
##   NUMBER             MOVIE          TYPE  TOTAL
## 1      1  A Beautiful Mind  Best Picture 170.71
## 2      2   American Beauty  Best Picture 130.06
## 3      3            Batman Biggest Gross 251.19
## 4      4 Beverly Hills Cop Biggest Gross 234.76
## 5      5           Chicago  Best Picture 170.69
## 6      6             Crash  Best Picture  55.33
## 7      7     Departed, The  Best Picture 133.31
```

```
summary(movie_weekend)
```

```
##      NUMBER         MOVIE              WEEK_NUM     WEEKEND_PER_THEATER WEEKEND_DATE
##  Min.   : 1.00   Length:1292        Min.   : 1.00   Min.   :  128       Length:1292
##  1st Qu.:11.00   Class :character   1st Qu.: 7.00   1st Qu.: 1034       Class :character
##  Median :25.00   Mode  :character   Median :13.00   Median : 1922       Mode  :character
##  Mean   :24.24                      Mean   :14.92   Mean   : 3635
##  3rd Qu.:36.00                      3rd Qu.:21.00   3rd Qu.: 3733
##  Max.   :49.00                      Max.   :52.00   Max.   :53846
##  NA's   :38                         NA's   :38      NA's   :38
```

```
summary(movie_daily)
```

```
##     NUMBER             MOVIE              DAY_NUM       DAILY_PER_THEATER      DATE
##  Length:2501        Length:2501        Min.   :  1.00   Length:2501        Length:2501
##  Class :character   Class :character   1st Qu.: 19.00   Class :character   Class :character
##  Mode  :character   Mode  :character   Median : 42.00   Mode  :character   Mode  :character
##                                        Mean   : 53.17
##                                        3rd Qu.: 75.00
```

```
##                                        Max.   :186.00
##                                        NA's   :47
```

```
summary(movie_total)
```

```
##      NUMBER        MOVIE               TYPE                TOTAL
##  Min.   : 1   Length:49          Length:49          Min.   :  1.28
##  1st Qu.:13   Class :character   Class :character   1st Qu.:100.32
##  Median :25   Mode  :character   Mode  :character   Median :261.99
##  Mean   :25                                         Mean   :228.55
##  3rd Qu.:37                                         3rd Qu.:321.01
##  Max.   :49                                         Max.   :600.79
```

Перетворення та очищення даних:

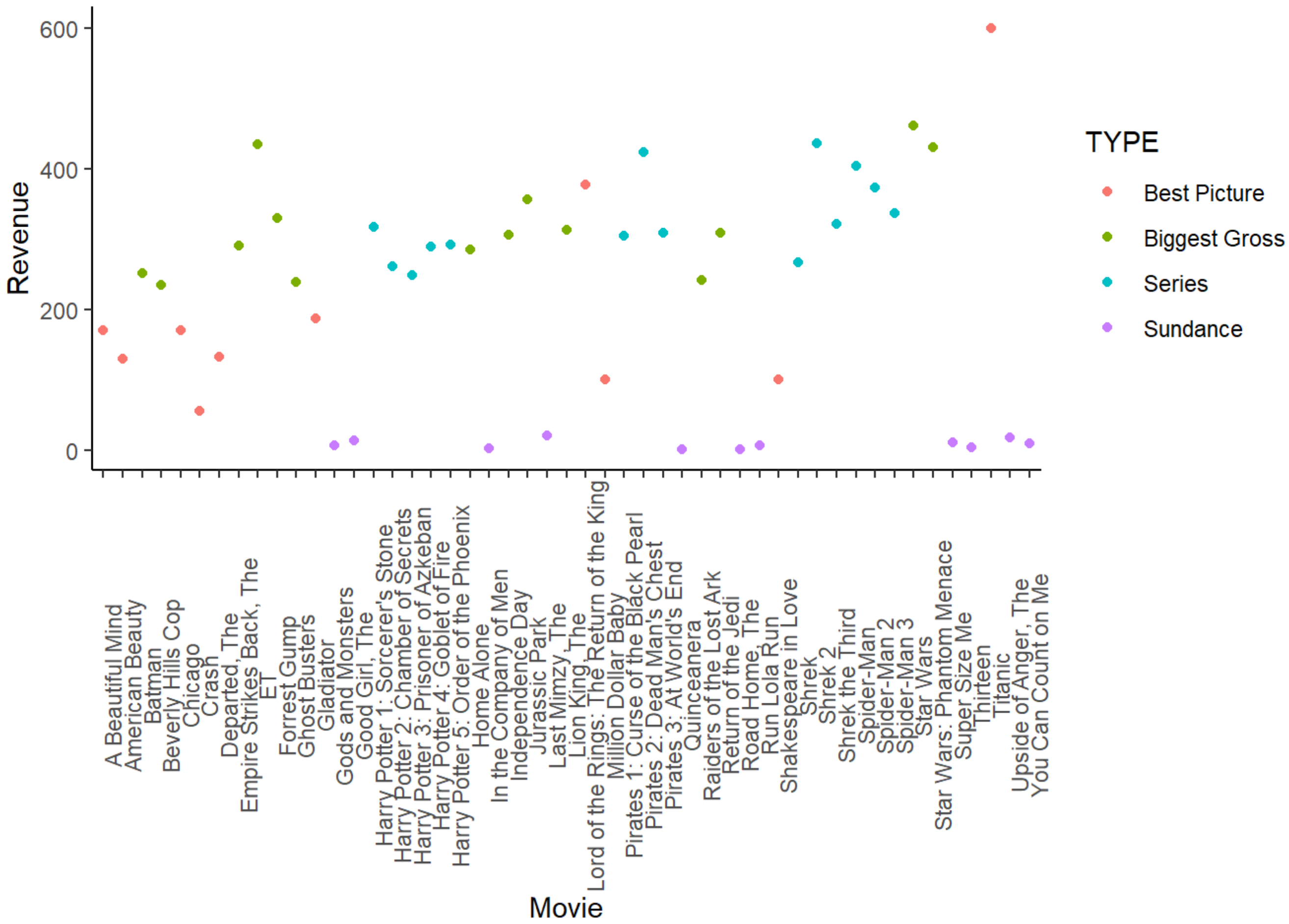
```
movie_weekend$WEEKEND_DATE <- as.Date.character(movie_weekend$WEEKEND_DATE, "%m/%d/%Y")
movie_weekend <- na.exclude(movie_weekend)
```

```
movie_daily[,c(1,4)] <- movie_daily[,c(1,4)] %>% sapply(as.integer)
movie_daily$DATE <- as.Date.character(movie_daily$DATE, "%m/%d/%Y")
movie_daily <- na.exclude(movie_daily)
head(movie_daily, 10)
```

```
##    NUMBER            MOVIE DAY_NUM DAILY_PER_THEATER       DATE
## 1       1 A Beautiful Mind       1              8909 2001-12-24
## 2       1 A Beautiful Mind       2              3885 2001-12-25
## 3       1 A Beautiful Mind       3              3365 2001-12-26
## 4       1 A Beautiful Mind       4              3324 2001-12-27
## 5       1 A Beautiful Mind       5              4403 2001-12-28
## 6       1 A Beautiful Mind       6              5475 2001-12-29
## 7       1 A Beautiful Mind       7              4964 2001-12-30
## 8       1 A Beautiful Mind       8              4126 2001-12-31
## 9       1 A Beautiful Mind       9              5110 2002-01-01
## 10      1 A Beautiful Mind      10              2606 2002-01-02
```

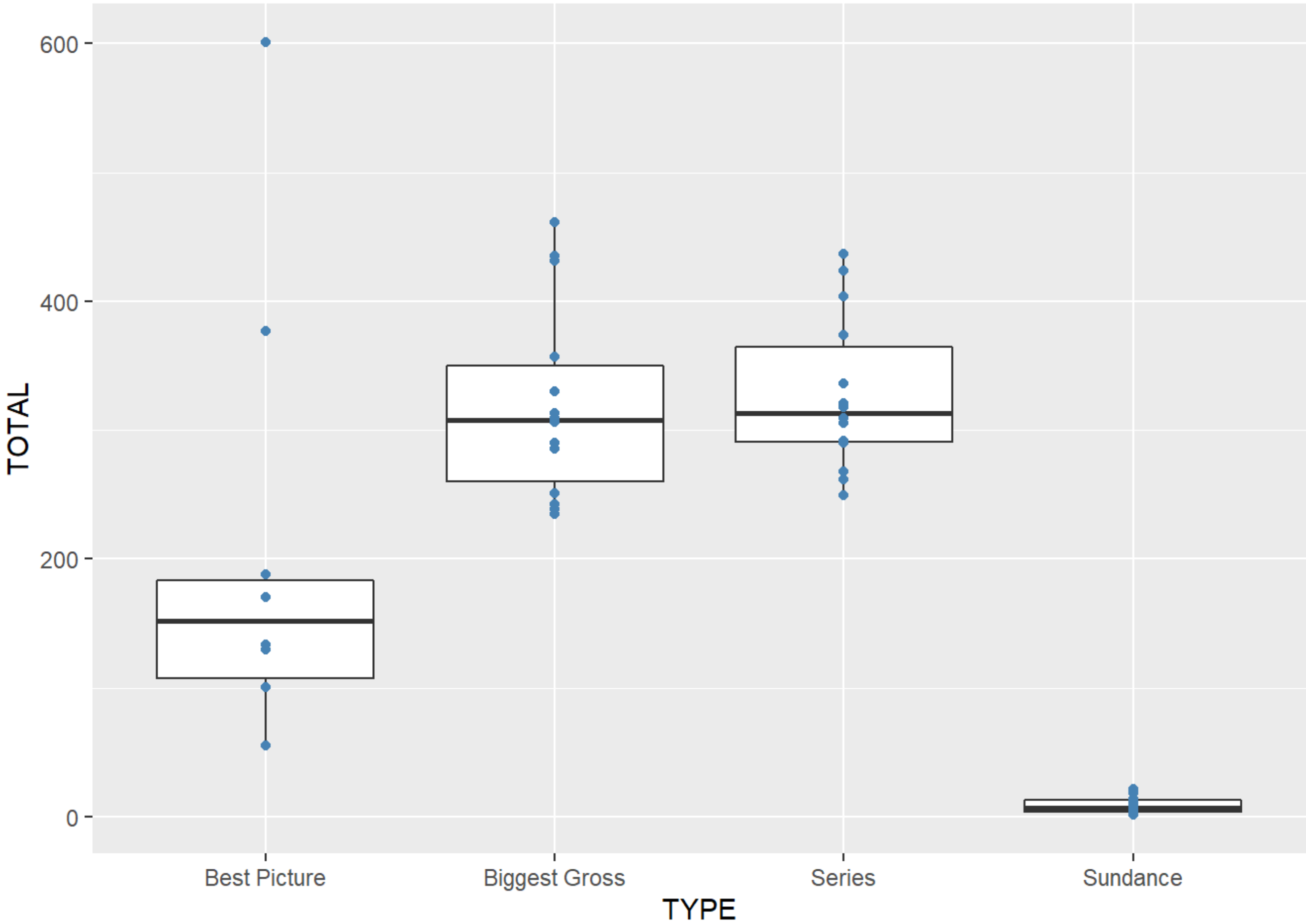
Розподіл кінофільмів за типами:

```
p1 <- ggplot(movie_total, aes(x = MOVIE, y = TOTAL, colour = TYPE)) +
  geom_point() +
  labs(x = "Movie", y = "Revenue") +
  scale_y_continuous(labels = comma) +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90))
p1
```



Boxplot кінокартин за типами:

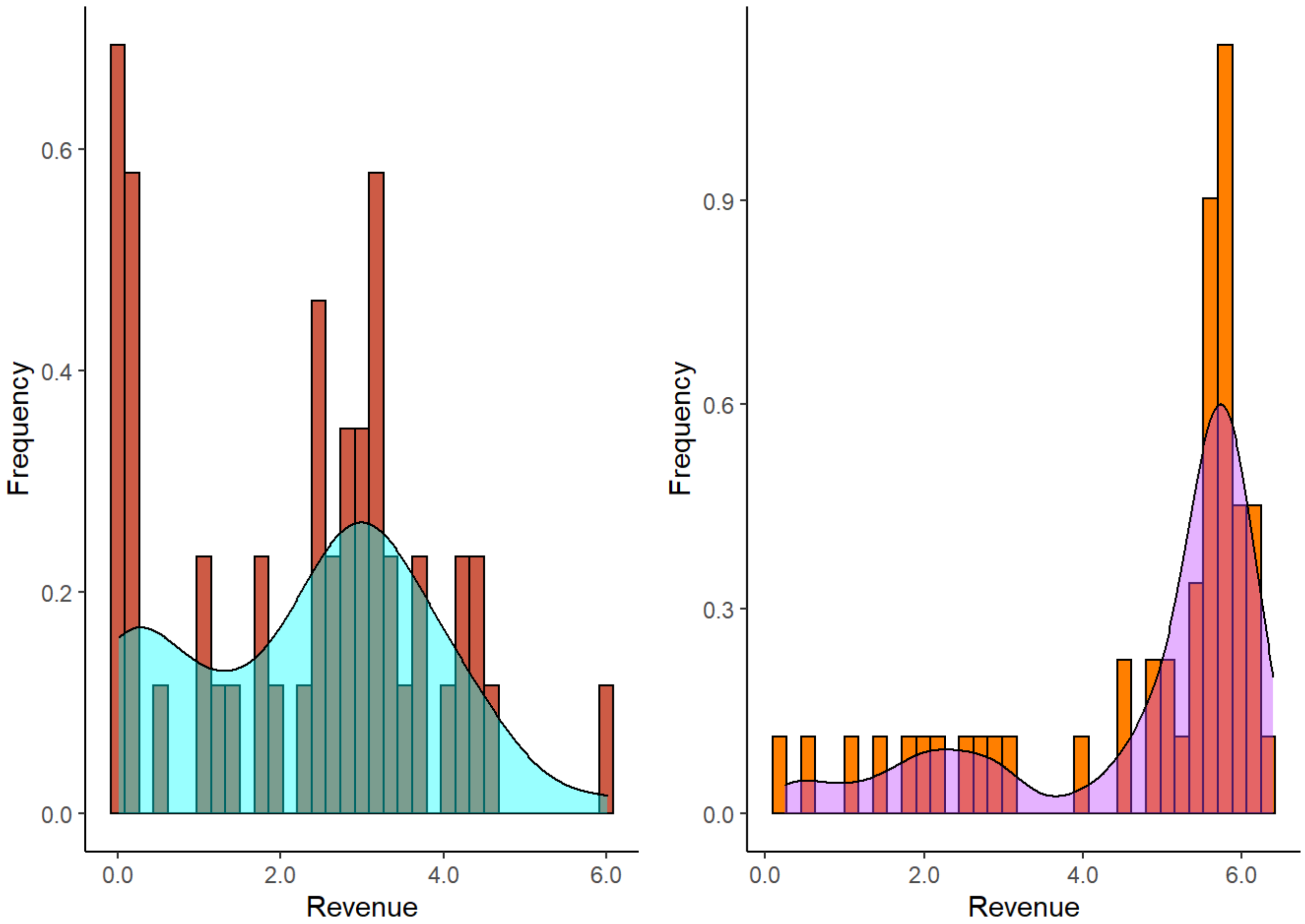
```
p2 <- ggplot(movie_total, aes(x = TYPE, y = TOTAL)) +
  geom_boxplot() +
  geom_point(color = 'steelblue')
p2
```



Гістограми (з і без scale):

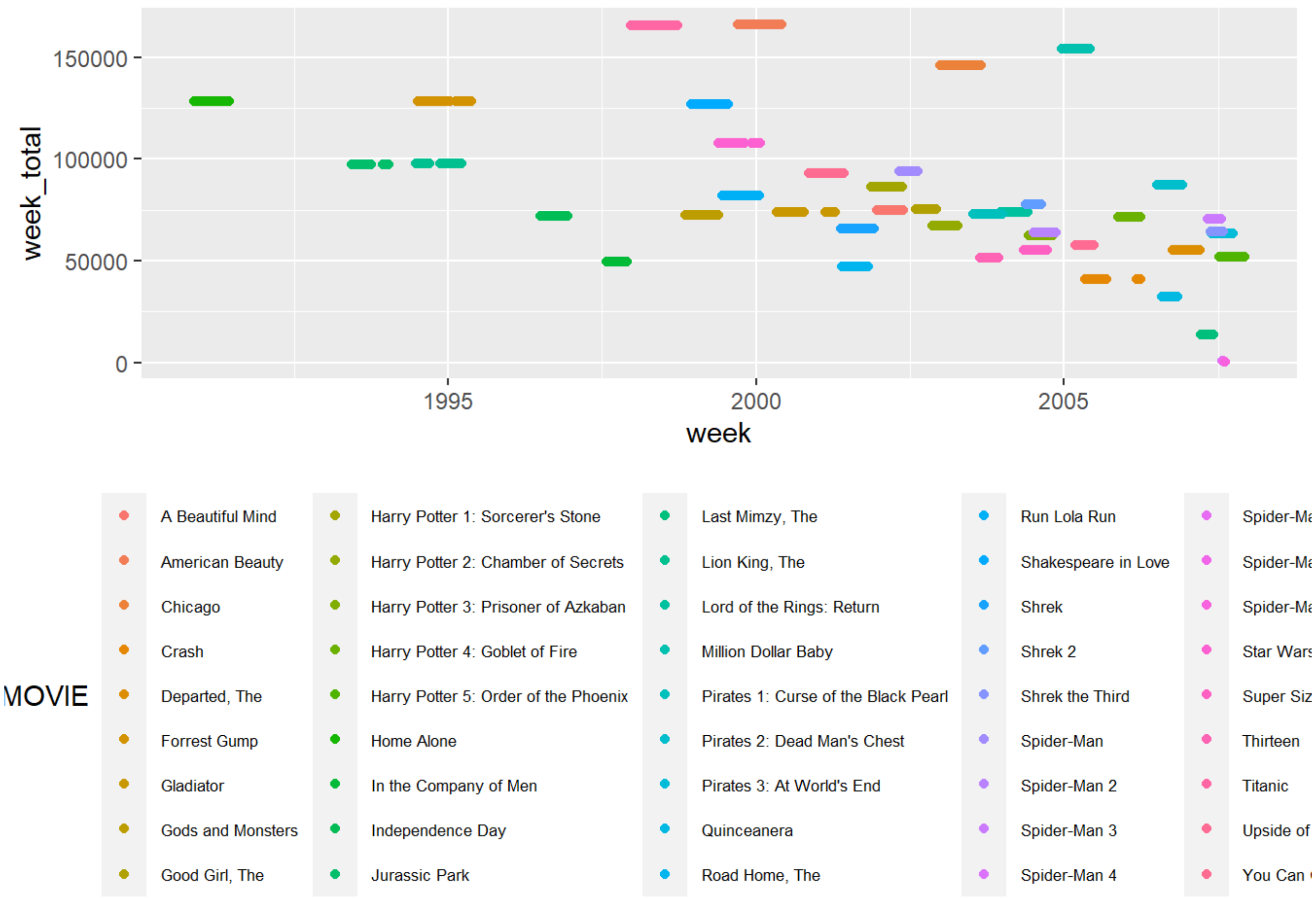
```
p1a <- ggplot(movie_total %>% filter(TOTAL != 0), aes(x = TOTAL/100)) +
  geom_histogram(aes(y=..density..), fill = "coral3", color = 'black', bins = 35) +
  geom_density(alpha=.4, fill="cyan1") +
  scale_x_continuous(labels = comma) +
  scale_y_continuous() +
```

```
  labs(x = "Revenue", y = "Frequency") +
  theme_classic()
p2a <- ggplot(movie_total %>% filter(TOTAL != 0), aes(x = log(TOTAL))) +
  geom_histogram(aes(y=..density..), fill = "darkorange1", color = 'black', bins = 35) +
  geom_density(alpha=.4, fill="darkorchid1") +
  scale_x_continuous(labels = comma) +
  scale_y_continuous() +
  labs(x = "Revenue", y = "Frequency") +
  theme_classic()
ggarrange(p1a, p2a)
```



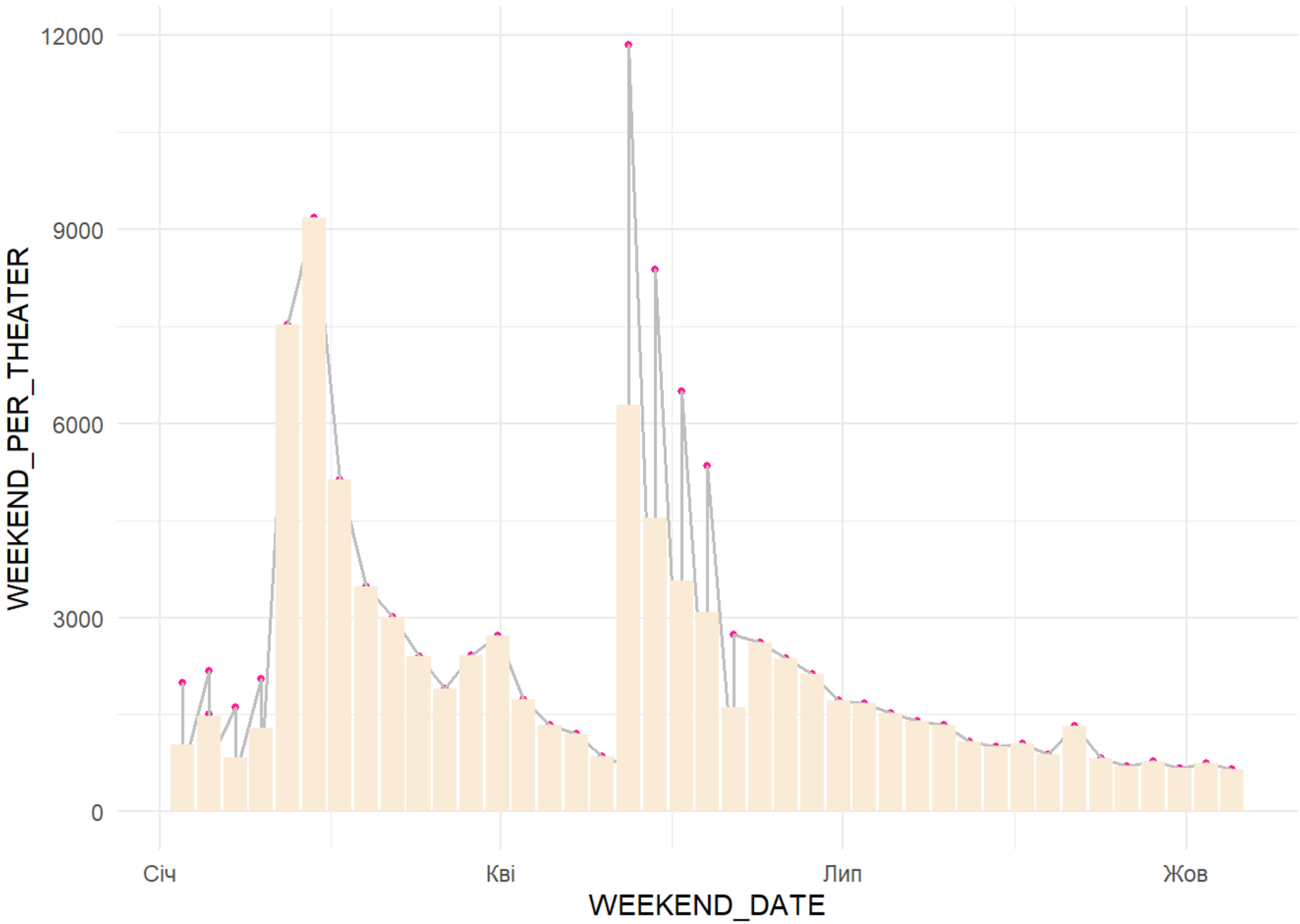

```
mV <- movie_weekend %>% filter(WEEKEND_DATE >= as.Date("1990-01-01")) %>% select(MOVIE, WEEKEND_PER_THEATER, WEEK
END_DATE) %>% group_by(MOVIE) %>%
  summarise(week_total = sum(WEEKEND_PER_THEATER), week = WEEKEND_DATE)
ggplot(mV, aes(x=week, y=week_total, colour = MOVIE)) +
  geom_point() +
  theme(legend.position = "bottom",
```

```
        legend.text = element_text(size = 6.5),
        legend.box.margin = margin(1, 1, 1, 5))
```




```
ggplot(movie_weekend %>% filter(WEEKEND_DATE >= as.Date("2000-01-01") & WEEKEND_DATE <= as.Date("2000-10-20")
                                & WEEKEND_PER_THEATER >= 200),  aes(x = WEEKEND_DATE, y = WEEKEND_PER_THEATER)) +
  geom_point(color = "deeppink1", size = 1) +
```

```
  geom_line(color = "grey", size = 0.7) +
  geom_bar(position = "dodge", stat = "summary", fun.y = "mean", fill = "antiquewhite", size = 2) +
  theme_minimal()
```



```
mw <- movie_weekend %>%
  filter(WEEKEND_DATE >= as.Date("2000-01-01")) %>%
  select(MOVIE, WEEKEND_DATE, WEEKEND_PER_THEATER) %>%
  group_by(MOVIE) %>%
  summarise(MOVIE, week_total = cumsum(WEEKEND_PER_THEATER), week = WEEKEND_DATE)
mw
```

```
## # A tibble: 656 x 3
## # Groups:   MOVIE [36]
##    MOVIE            week_total week
##    <chr>                 <int> <date>
##  1 A Beautiful Mind        701 2001-12-21
##  2 A Beautiful Mind      15521 2001-12-28
##  3 A Beautiful Mind      24461 2002-01-04
##  4 A Beautiful Mind      31311 2002-01-11
##  5 A Beautiful Mind      36591 2002-01-18
##  6 A Beautiful Mind      41746 2002-01-25
##  7 A Beautiful Mind      45481 2002-02-01
##  8 A Beautiful Mind      48321 2002-02-08
##  9 A Beautiful Mind      52211 2002-02-15
## 10 A Beautiful Mind      54776 2002-02-22
## # ... with 646 more rows
```

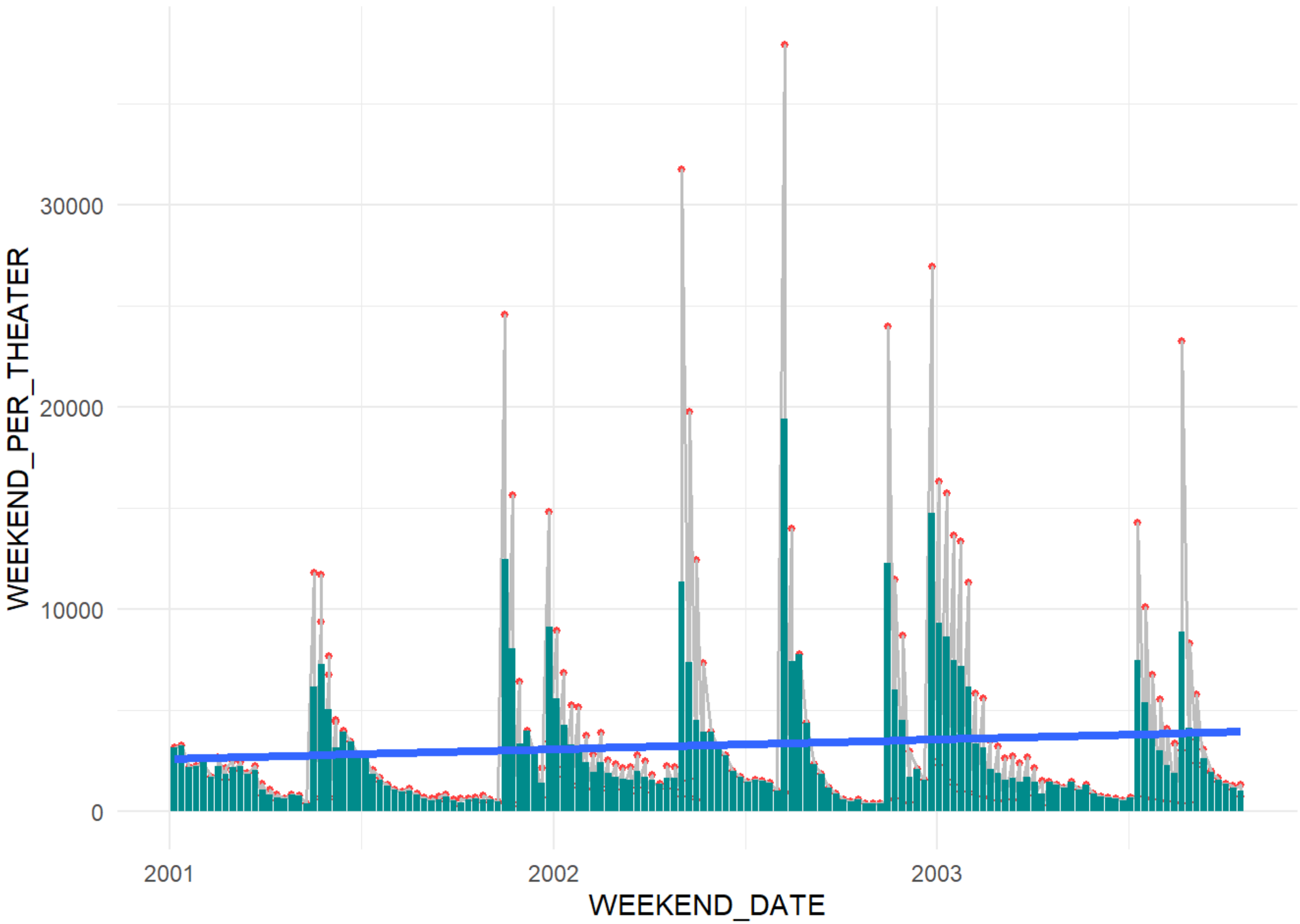
```
movie_weekend %>% filter(WEEKEND_DATE >= as.Date("2000-01-01") & WEEKEND_DATE <= as.Date("2000-12-12")
                                & WEEKEND_PER_THEATER >= 200)
```

```
##    NUMBER                     MOVIE WEEK_NUM WEEKEND_PER_THEATER WEEKEND_DATE
## 1       2           American Beauty       17                1998   2000-01-07
## 2       2           American Beauty       18                2168   2000-01-14
## 3       2           American Beauty       19                1614   2000-01-21
```

```
## 4        2           American Beauty     20                2047   2000-01-28
## 5        2           American Beauty     21                7523   2000-02-04
## 6        2           American Beauty     22                9185   2000-02-11
## 7        2           American Beauty     23                5132   2000-02-18
## 8        2           American Beauty     24                3483   2000-02-25
## 9        2           American Beauty     25                3010   2000-03-03
## 10       2           American Beauty     26                2408   2000-03-10
## 11       2           American Beauty     27                1902   2000-03-17
## 12       2           American Beauty     28                2422   2000-03-24
## 13       2           American Beauty     29                2717   2000-03-31
## 14       2           American Beauty     30                1731   2000-04-07
## 15       2           American Beauty     31                1341   2000-04-14
## 16       2           American Beauty     32                1201   2000-04-21
## 17       2           American Beauty     33                 848   2000-04-28
## 18       2           American Beauty     34                 711   2000-05-05
## 19       2           American Beauty     35                 724   2000-05-12
## 20       2           American Beauty     36                 640   2000-05-19
## 21       2           American Beauty     37                 827   2000-05-26
## 22       2           American Beauty     38                 479   2000-06-02
## 23      12                 Gladiator      1               11851   2000-05-05
## 24      12                 Gladiator      2                8374   2000-05-12
## 25      12                 Gladiator      3                6494   2000-05-19
## 26      12                 Gladiator      4                5353   2000-05-26
## 27      12                 Gladiator      5                2741   2000-06-02
## 28      12                 Gladiator      6                2614   2000-06-09
## 29      12                 Gladiator      7                2366   2000-06-16
## 30      12                 Gladiator      8                2126   2000-06-23
## 31      12                 Gladiator      9                1720   2000-06-30
## 32      12                 Gladiator     10                1677   2000-07-07
## 33      12                 Gladiator     11                1520   2000-07-14
## 34      12                 Gladiator     12                1408   2000-07-21
## 35      12                 Gladiator     13                1343   2000-07-28
## 36      12                 Gladiator     14                1090   2000-08-04
## 37      12                 Gladiator     15                1014   2000-08-11
## 38      12                 Gladiator     16                1045   2000-08-18
## 39      12                 Gladiator     17                 889   2000-08-25
## 40      12                 Gladiator     18                1325   2000-09-01
```

```
## 41      12                 Gladiator       19                  828   2000-09-08
## 42      12                 Gladiator       20                  703   2000-09-15
## 43      12                 Gladiator       21                  782   2000-09-22
## 44      12                 Gladiator       22                  672   2000-09-29
## 45      12                 Gladiator       23                  742   2000-10-06
## 46      12                 Gladiator       24                  663   2000-10-13
## 47      35              Run Lola Run       29                  497   2000-01-07
## 48      35              Run Lola Run       30                 1505   2000-01-14
## 49      35              Run Lola Run       31                  364   2000-01-21
## 50      44 Star Wars: Phantom Menace       28                  611   2000-01-07
## 51      44 Star Wars: Phantom Menace       29                  766   2000-01-14
## 52      44 Star Wars: Phantom Menace       30                  557   2000-01-21
## 53      44 Star Wars: Phantom Menace       31                  558   2000-01-28
## 54      49        You Can Count on Me       1                14771   2000-11-10
## 55      49        You Can Count on Me       2                 8614   2000-11-17
## 56      49        You Can Count on Me       3                 9708   2000-11-24
## 57      49        You Can Count on Me       4                 7368   2000-12-01
## 58      49        You Can Count on Me       5                 5607   2000-12-08
```

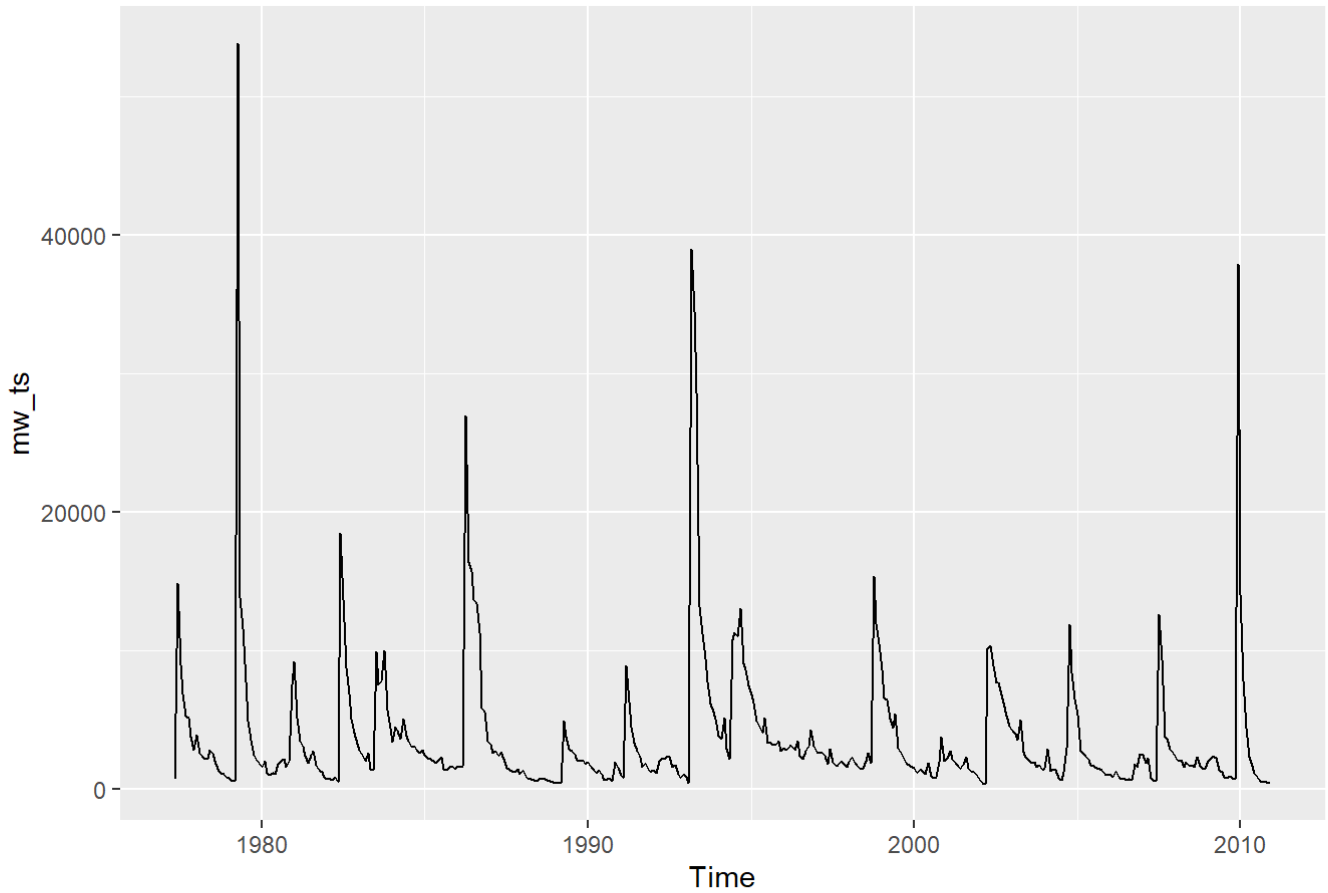
```
ggplot(movie_weekend %>% filter(WEEKEND_DATE >= as.Date("2001-01-01") & WEEKEND_DATE <= as.Date("2003-10-20")),
 aes(x = WEEKEND_DATE, y = WEEKEND_PER_THEATER)) +
  geom_point(color = "brown1", size = 1) +
  geom_line(color = "grey", size = 0.7) +
  geom_bar(position = "dodge", stat = "summary", fun.y = "mean", fill = "cyan4", size = 2) +
  geom_smooth(method="lm", se=FALSE, size = 1.5) +
  theme_minimal()
```




```
mw_ts <- ts(movie_weekend$WEEKEND_PER_THEATER, frequency=12, start = c(1977, 05), end = c(2010, 12))
str(mw_ts)
```

```
##  Time-Series [1:404] from 1977 to 2011: 701 14820 8940 6850 5280 5155 3735 2840 3890 2565 ...
```

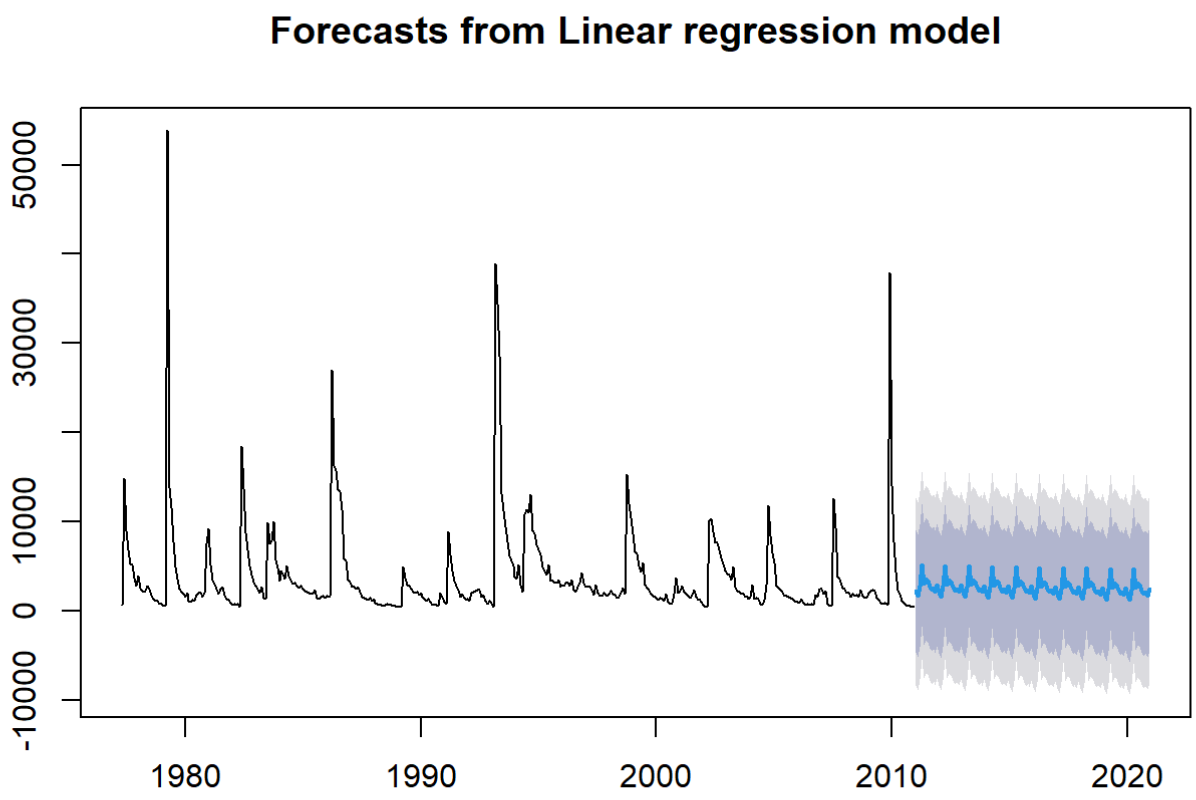
```
autoplot(mw_ts)
```




```
df_ts <- data.frame(revenue = mw_ts, as.numeric(time(mw_ts)))
names(df_ts) <- c("revenue", "time")
fit.consMR <- tslm(
  revenue ~ season + trend - 1,
  data=df_ts)
summary(fit.consMR)
```

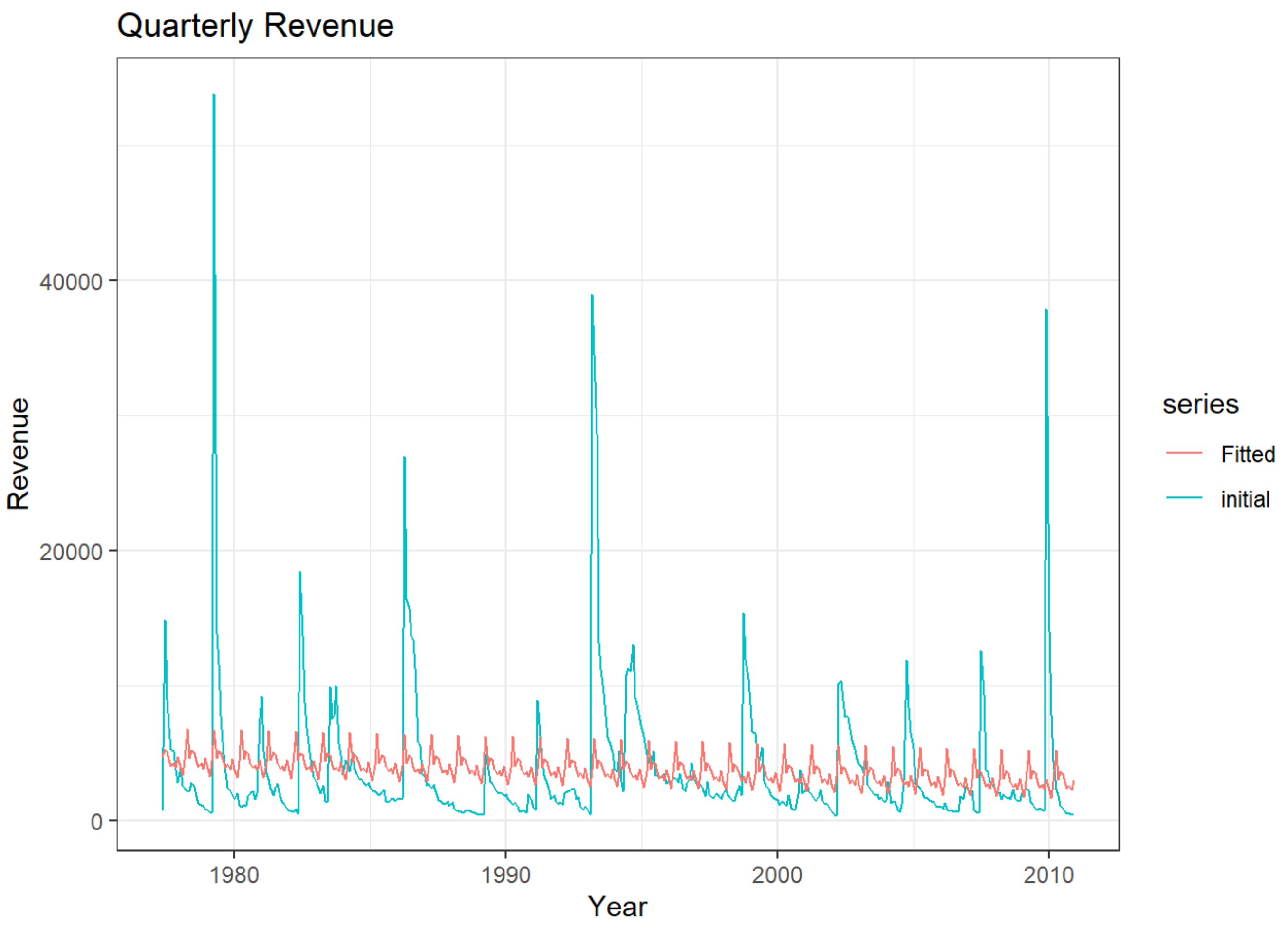
```
##
## Call:
## tslm(formula = revenue ~ season + trend - 1, data = df_ts)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -5769  -2416  -1365    319  47098
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## season1  3904.590   1018.457   3.834 0.000147 ***
## season2  3349.409   1019.448   3.286 0.001110 **
## season3  4531.531   1020.444   4.441 1.17e-05 ***
## season4  6848.744   1021.443   6.705 7.07e-11 ***
## season5  4708.995   1004.337   4.689 3.81e-06 ***
## season6  5277.708   1005.333   5.250 2.51e-07 ***
## season7  5080.833   1006.332   5.049 6.83e-07 ***
## season8  4467.517   1007.336   4.435 1.20e-05 ***
## season9  4058.259   1008.343   4.025 6.85e-05 ***
## season10 4309.531   1009.355   4.270 2.46e-05 ***
## season11 3922.862   1010.370   3.883 0.000121 ***
## season12 4698.664   1011.390   4.646 4.64e-06 ***
## trend      -4.213      2.239  -1.882 0.060639 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5248 on 391 degrees of freedom
## Multiple R-squared:  0.3595, Adjusted R-squared:  0.3382
## F-statistic: 16.88 on 13 and 391 DF,  p-value: < 2.2e-16
```

```
my_fc <- forecast(fit.consMR, h=120)
plot(my_fc)
```

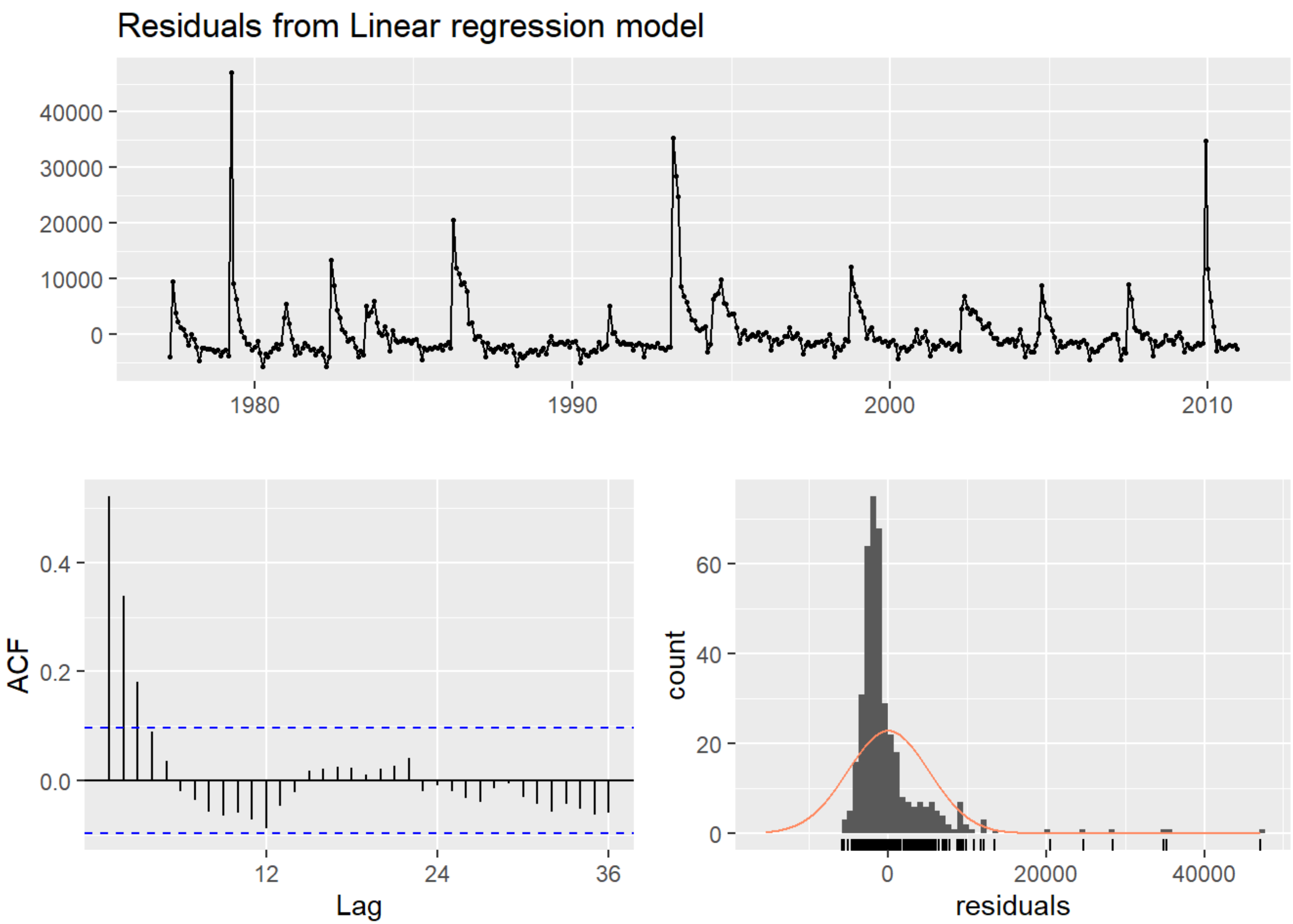



```
autoplot(mw_ts, series="initial") +
  autolayer(fitted(fit.consMR), series="Fitted") +
```

```
  xlab("Year") + ylab("Revenue") +
  ggtitle("Quarterly Revenue") +
  theme_bw()
```



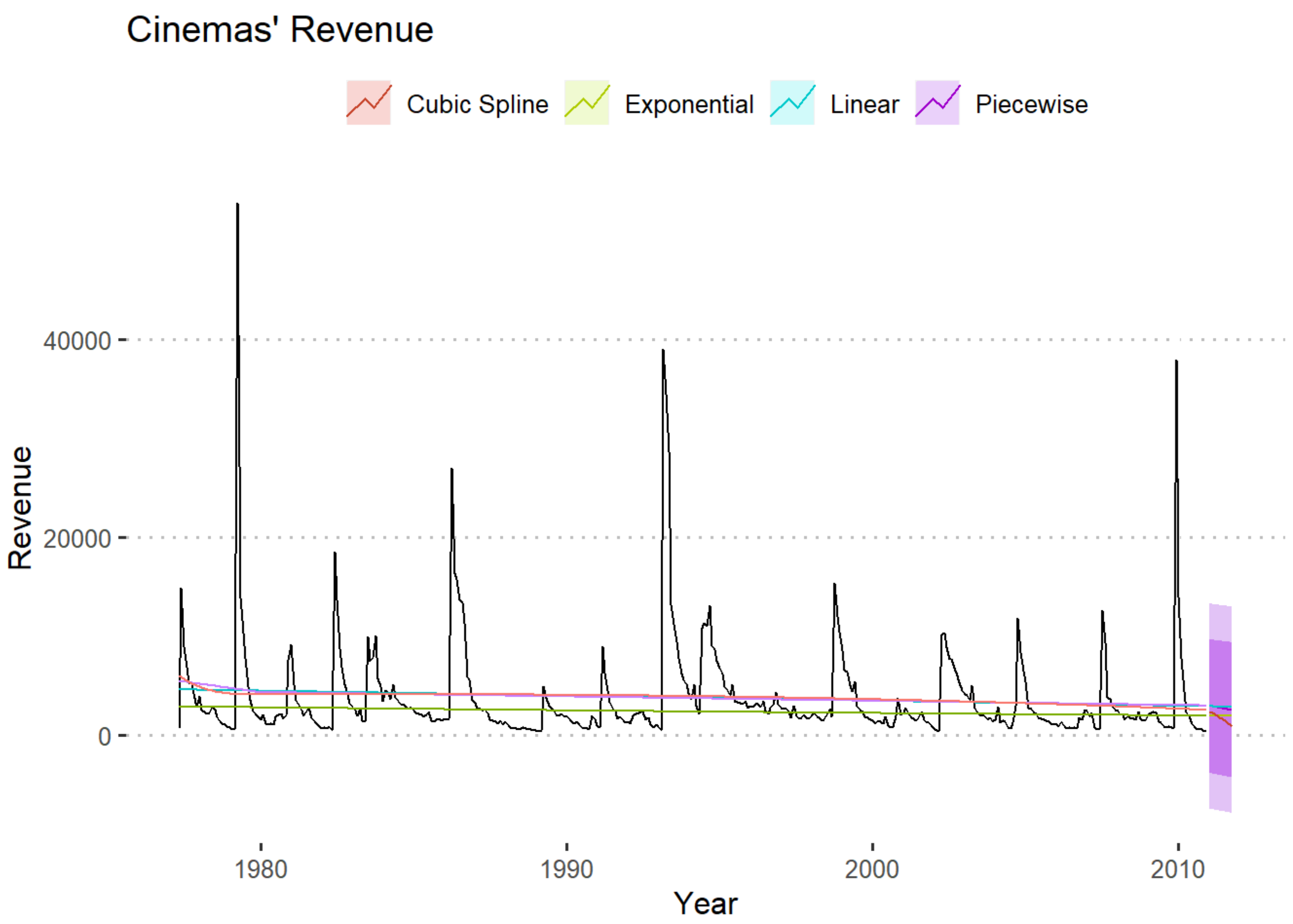

```
checkresiduals(fit.consMR)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 24
##
## data:  Residuals from Linear regression model
## LM test = 118.46, df = 24, p-value = 1.822e-14
```

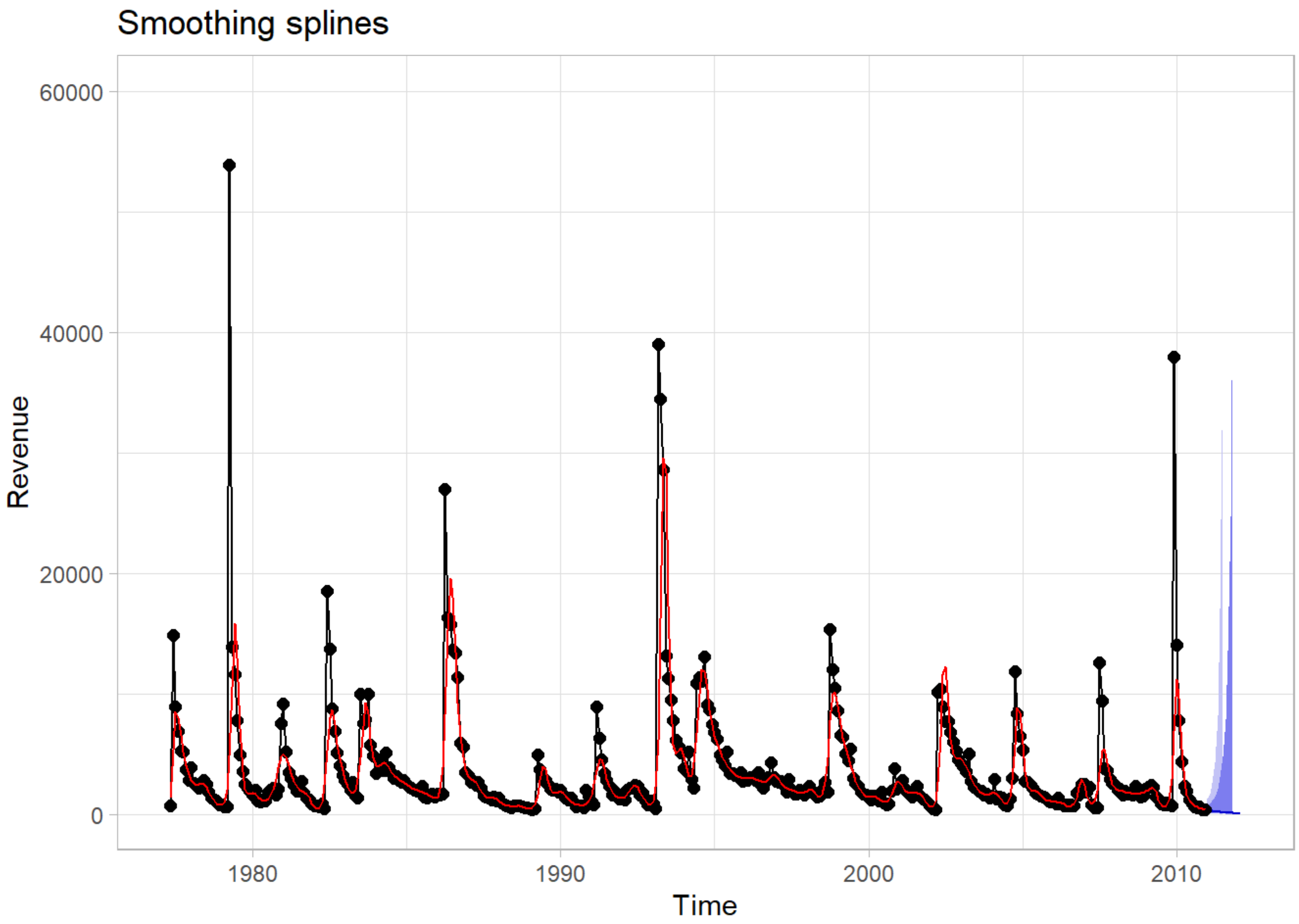
```
h <- 10
fit.lin <- tslm(revenue ~ trend,data=df_ts)
fcasts.lin <- forecast(fit.lin, h = h)
fit.exp <- tslm(revenue ~ trend, data=df_ts, lambda = 0)
fcasts.exp <- forecast(fit.exp, h = h)

t <- time(mw_ts)
t.break1 <- 1970
t.break2 <- 1980
tb1 <- ts(pmax(0, t - t.break1), start = 1977)
tb2 <- ts(pmax(0, t - t.break2), start = 2008)

fit.pw <- tslm(revenue ~ t + tb1 + tb2, data=df_ts)
t.new <- t[length(t)] + seq(h)
tb1.new <- tb1[length(tb1)] + seq(h)
tb2.new <- tb2[length(tb2)] + seq(h)

newdata <- cbind(t=t.new, tb1=tb1.new, tb2=tb2.new) %>%
  as.data.frame()
fcasts.pw <- forecast(fit.pw, newdata = newdata)

fit.spline <- tslm(revenue ~ t + I(t^2) + I(t^3) +
  I(tb1^3) + I(tb2^3), data=df_ts)
fcasts.spl <- forecast(fit.spline, newdata = newdata)

autoplot(mw_ts) +
  autolayer(fitted(fit.lin), series = "Linear") +
  autolayer(fitted(fit.exp), series = "Exponential") +
  autolayer(fitted(fit.pw), series = "Piecewise") +
  autolayer(fitted(fit.spline), series = "Cubic Spline") +
  autolayer(fcasts.pw, series="Piecewise") +
  autolayer(fcasts.lin, series="Linear", PI=FALSE) +
  autolayer(fcasts.exp, series="Exponential", PI=FALSE) +
  autolayer(fcasts.spl, series="Cubic Spline", PI=FALSE) +
  xlab("Year") + ylab("Revenue") +
  ggtitle("Cinemas' Revenue") +
```

```
  guides(colour = guide_legend(title = " ")) +
  theme_pubclean()
```



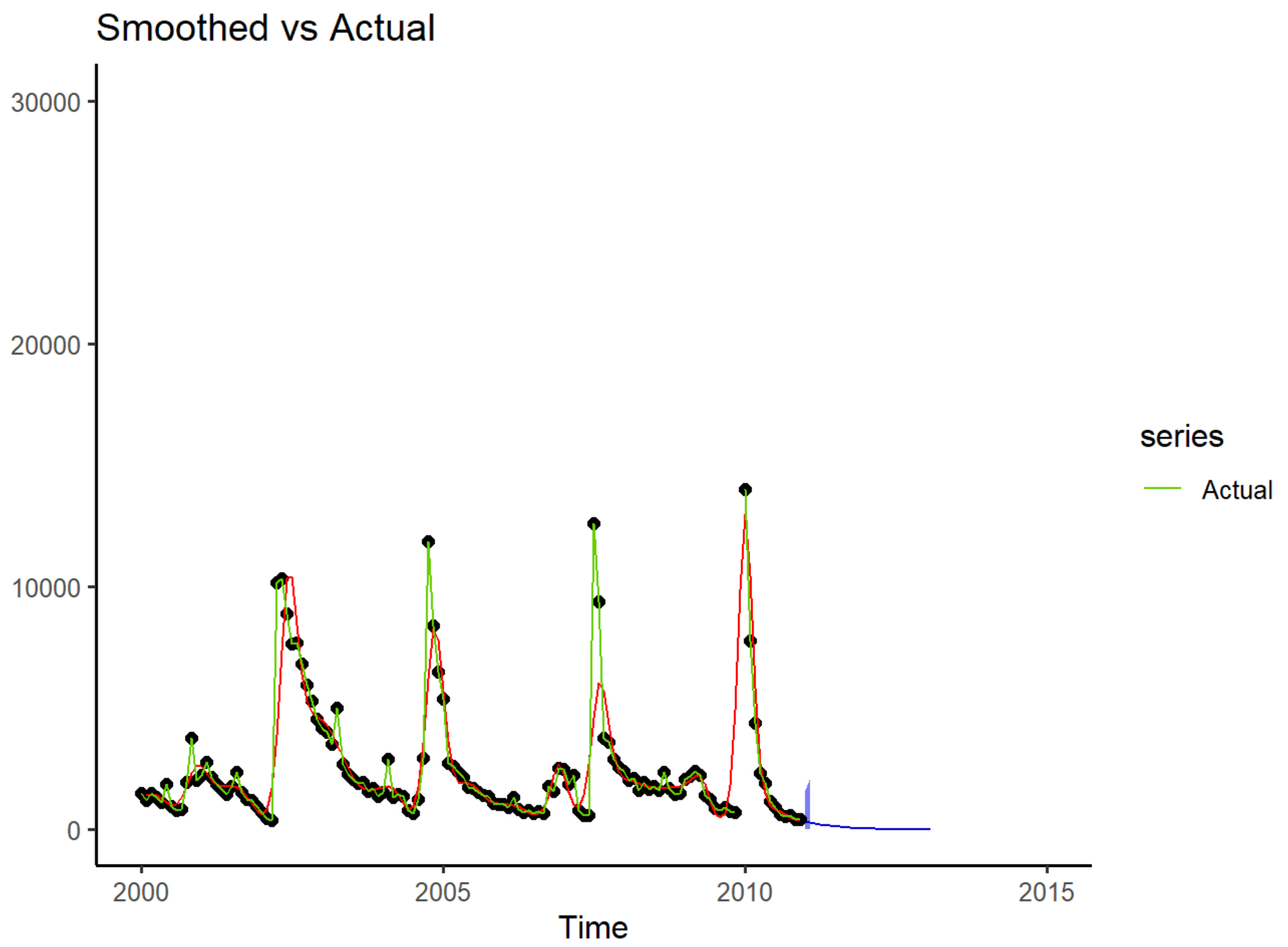

```
splinef(mw_ts, h=14, lambda=-0.05, method = "gcv") %>%
  autoplot(ylim = c(0, 60000)) +
  ylab("Revenue") +
```

```
  ggtitle("Smoothing splines") +
  theme_light()
```




```
spl_pred <- predict(mw_ts %>% splinef(h=300, lambda=0.3), mw_ts)
autoplot(spl_pred, ylim = c(0, 30000), xlim = c(2000, 2015)) +
autolayer(mw_ts) +
  scale_color_manual(labels = c("Actual", "Forecasted"),
```

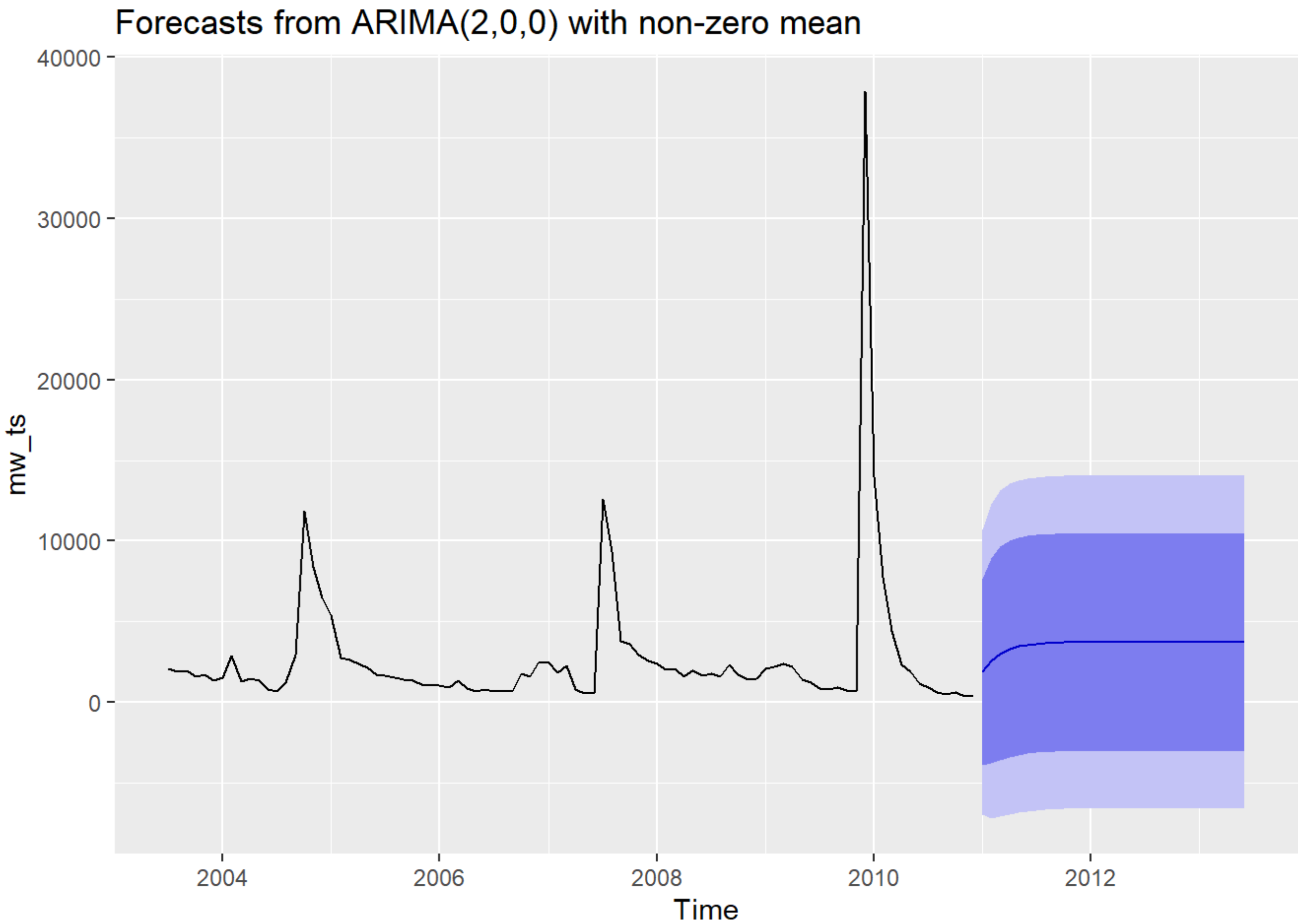
```
                  values=c("chartreuse3", "darkred3")) +
  ggtitle("Smoothed vs Actual") +
  theme_classic2()
```




```
spl_model <- splinef(mw_ts, h=300, lambda=-0.05, method = "gcv")
accuracy(spl_pred)
```

```
##                        ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
## Training set -413.4282 6147.484 2264.153 -12.91916 38.25337 0.5645504 0.2593321
```

```
fit <- auto.arima(mw_ts, seasonal=TRUE)
fit %>% forecast(h=30) %>% autoplot(include=90)
```



Random Forest

```
model_data_tbl <-
  movie_weekend %>%
  mutate(trend       = 1:nrow(movie_weekend),
         trend_sqr   = trend^2,
         rev_lag_13  = lag(WEEKEND_PER_THEATER, n = 13),
         rev_lag_52  = lag(WEEKEND_PER_THEATER, n = 52),
         season      = case_when(WEEKEND_PER_THEATER == 0 ~ 0,
                                 TRUE ~ 1)
        ) %>%
 filter(!is.na(rev_lag_52))

train_tbl <-
  model_data_tbl %>%
  filter(WEEKEND_DATE <= "2007-03-19")

test_tbl <-
  model_data_tbl %>%
  filter(WEEKEND_DATE >= "2006-10-02" &
           WEEKEND_DATE <= "2007-03-19")

train_tbl %>% head()
```

```
##   NUMBER           MOVIE WEEK_NUM WEEKEND_PER_THEATER WEEKEND_DATE trend trend_sqr rev_lag_13 rev_lag_52 seaso
n
## 1      2 American Beauty       30                1731   2000-04-07    53      2809       1998        701
1
## 2      2 American Beauty       31                1341   2000-04-14    54      2916       2168      14820
1
## 3      2 American Beauty       32                1201   2000-04-21    55      3025       1614       8940
1
```

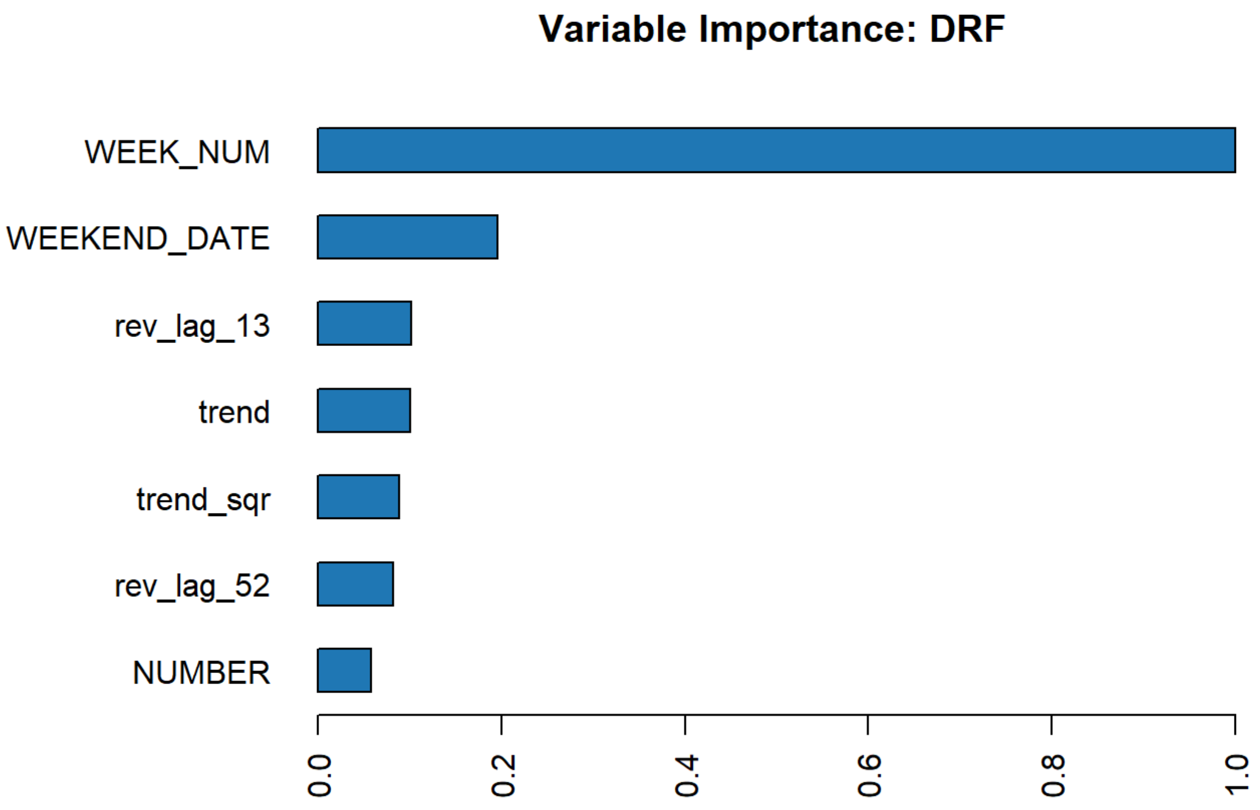
```
## 4      2 American Beauty       33                  848   2000-04-28    56      3136       2047       6850
1
## 5      2 American Beauty       34                  711   2000-05-05    57      3249       7523       5280
1
## 6      2 American Beauty       35                  724   2000-05-12    58      3364       9185       5155
1
```

```
h2o.init(max_mem_size = "8G")
```

```
##  Connection successful!
##
## R is connected to the H2O cluster:
##     H2O cluster uptime:         6 minutes 29 seconds
##     H2O cluster timezone:       Europe/Helsinki
##     H2O data parsing timezone:  UTC
##     H2O cluster version:        3.32.0.1
##     H2O cluster version age:    3 months and 1 day
##     H2O cluster name:           H2O_started_from_R_danie_xim214
##     H2O cluster total nodes:    1
##     H2O cluster total memory:   6.93 GB
##     H2O cluster total cores:    8
##     H2O cluster allowed cores:  8
##     H2O cluster healthy:        TRUE
##     H2O Connection ip:          localhost
##     H2O Connection port:        54321
##     H2O Connection proxy:       NA
##     H2O Internal Security:      FALSE
##     H2O API Extensions:         Amazon S3, Algos, AutoML, Core V3, TargetEncoder, Core V4
##     R Version:                  R version 4.0.2 (2020-06-22)
```

```
h2o.no_progress()
y <- "WEEKEND_PER_THEATER"
```

```
# predictors set: remove response variable and order_date from the set
x <- setdiff(names(train_tbl %>% as.h2o()), c(y, "weekend_date"))

rft_model <-
  h2o.randomForest(
    x = x,
    y = y,
    training_frame = train_tbl %>% as.h2o(),
    nfolds = 10,
    ntrees = 500,
    stopping_metric = "RMSE",
    stopping_rounds = 10,
    stopping_tolerance = 0.005,
    seed = 1975
  )

rft_model %>% h2o.varimp_plot()
```




```
rft_model@model$model_summary
```

```
## Model Summary:
##   number_of_trees number_of_internal_trees model_size_in_bytes min_depth max_depth mean_depth min_leaves
## 1              42                       42              333989        18        20   19.88095        574
```

```
##   max_leaves mean_leaves
## 1        660   628.61900
```

```
h2o.performance(rft_model, newdata = test_tbl %>% as.h2o())
```

```
## H2ORegressionMetrics: drf
##
## MSE:  178418.9
## RMSE:  422.3966
## MAE:  236.2163
## RMSLE:  0.3098941
## Mean Residual Deviance :  178418.9
```

```
rft_model %>% h2o.r2()
```

```
## [1] 0.7992146
```

Naive Bayes

```
splice <- h2o.uploadFile("C:/Users/danie/Downloads/movietotal-dat.txt", header = TRUE, na.strings = FALSE)

# Set the predictors and response; set the response as a factor:
splice$TYPE <- as.factor(splice$TYPE)
predictors <- c("MOVIE", "TOTAL")
response <- "TYPE"

# Build and train the model:
pros_nb <- h2o.naiveBayes(x = predictors,
                          y = response,
                          training_frame = splice,
```

```r
                          laplace = 0,
                          nfolds = 5,
                          seed = 1234)

perf <- h2o.performance(pros_nb)
pred <- h2o.predict(pros_nb, newdata = splice)
pros_nb %>% h2o.r2()
```

```
## [1] 0.7761132
```