МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Кафедра

інформаційних систем

та мереж

Контрольна розрахункова робота №2

з дисципліни “***Методи обчислень***”

Варіант № 20

Виконали студенти групи СА-32

Іванчишин Д.Я., Волошинський О.Б.,

Кожомін А.М.

Прийняла доцент, к.т.н.

Висоцька В.А.

*Львів - 2020*

# **Вступ**

Дана розрахункова робота присвячена вивченню основних методів аналізу статистичних даних, які найчастіше застосовуються в наукових дослідженнях та на практиці, а також формуванню практичних навичок застосування цих методів за допомогою комп’ютерних програм аналізу даних.

Завдяки даній роботі було сформовано:

* оволодіння сучасними методами аналізу статистичних даних;
* здатність до критичного аналізу та оцінки сучасних наукових досягнень;
* вміння самостійно ставити задачу дослідження та забезпечувати її розв’язання;
* здатність проводити комп’ютерний аналіз даних;
* здатність аналізувати результати досліджень та забезпечувати їх застосування;
* вміння формулювати висновки та рекомендації.

***Метою даної роботи*** є ознайомлення з основними методами висвітлення тенденції поведінки досліджуваного показника, яка представлена характером його тренду, з допомогою методів згладжування часових рядів та подання отриманих результатів. Метою роботи також є ознайомлення з методами кореляційного аналізу експериментальних даних, поданих часовими послідовностями. Для цього потрібно пройти такі етапи:

1. Графічне подання і опис поведінки часового ряду.
2. Виділення та видалення закономірних складових часового ряду, що залежать від часу: [тренда](https://uk.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%BD%D0%B4_(%D1%81%D1%82%D0%B0%D1%82%D0%B8%D1%81%D1%82%D0%B8%D0%BA%D0%B0)), сезонних і циклічних складових.
3. Виділення та видалення низько- або високочастотних складових процесу (фільтрація).
4. Дослідження випадкової складової часового ряду, що залишилася після видалення перерахованих вище складових.
5. Побудова (підбір) математичної моделі для опису випадкової складової і перевірка її адекватності.
6. Прогнозування майбутнього розвитку процесу, представленого часовим рядом.

## 1. Попередня обробка даних та подання її результатів

### 1.1. Формування файлу даних у формі excel та word таблиць.

 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)

### 1.2. Попереднє очищення даних

  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

## 2. Описова статистика - кількісні характеристики даних.

### 2.1. Основні характеристики даних

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

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

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

### 2.2. Графічне подання даних.

  Розподіл кінофільмів за типами:

p1 <- ggplot(movie\_total, aes(x = MOVIE, y = TOTAL, colour = TYPE)) +

geom\_point() +

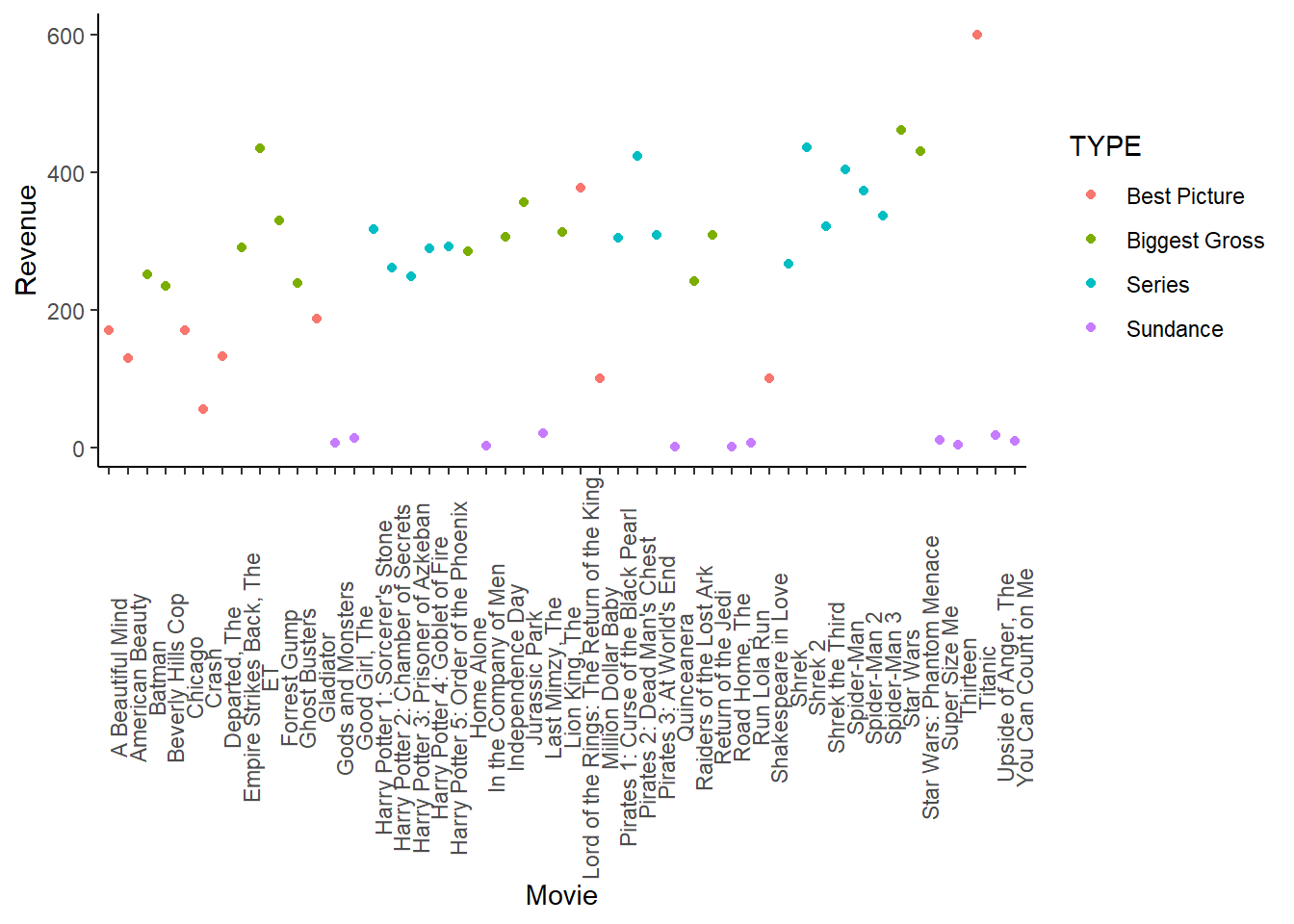
labs(x = "Movie", y = "Revenue") +

scale\_y\_continuous(labels = comma) +

theme\_classic() +

theme(axis.text.x = element\_text(angle = 90))

p1



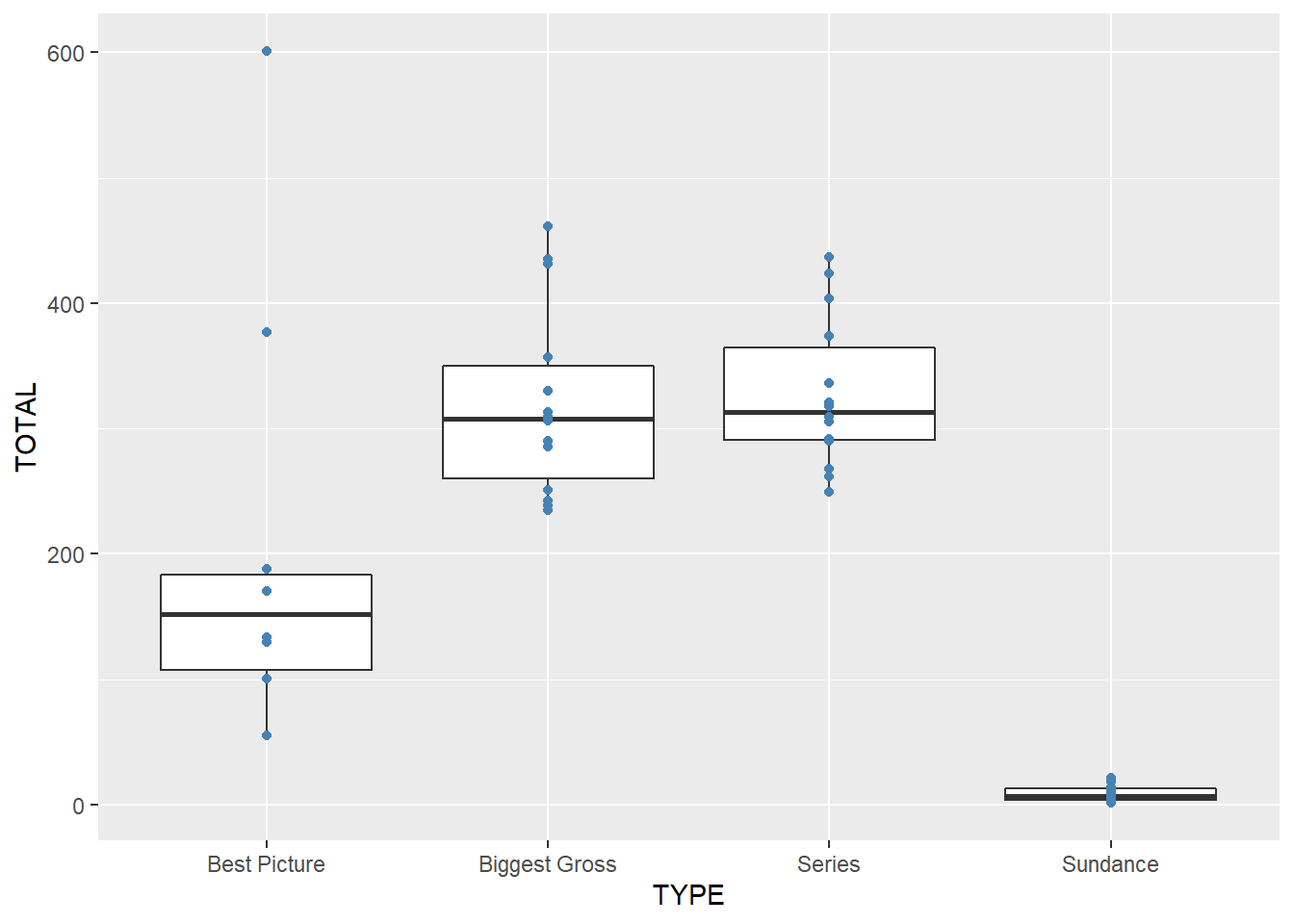
### 2.3. Bоxplot кінокартин за типами:

p2 <- ggplot(movie\_total, aes(x = TYPE, y = TOTAL)) +

geom\_boxplot() +

geom\_point(color = 'steelblue')

p2



### 2.4. 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") +

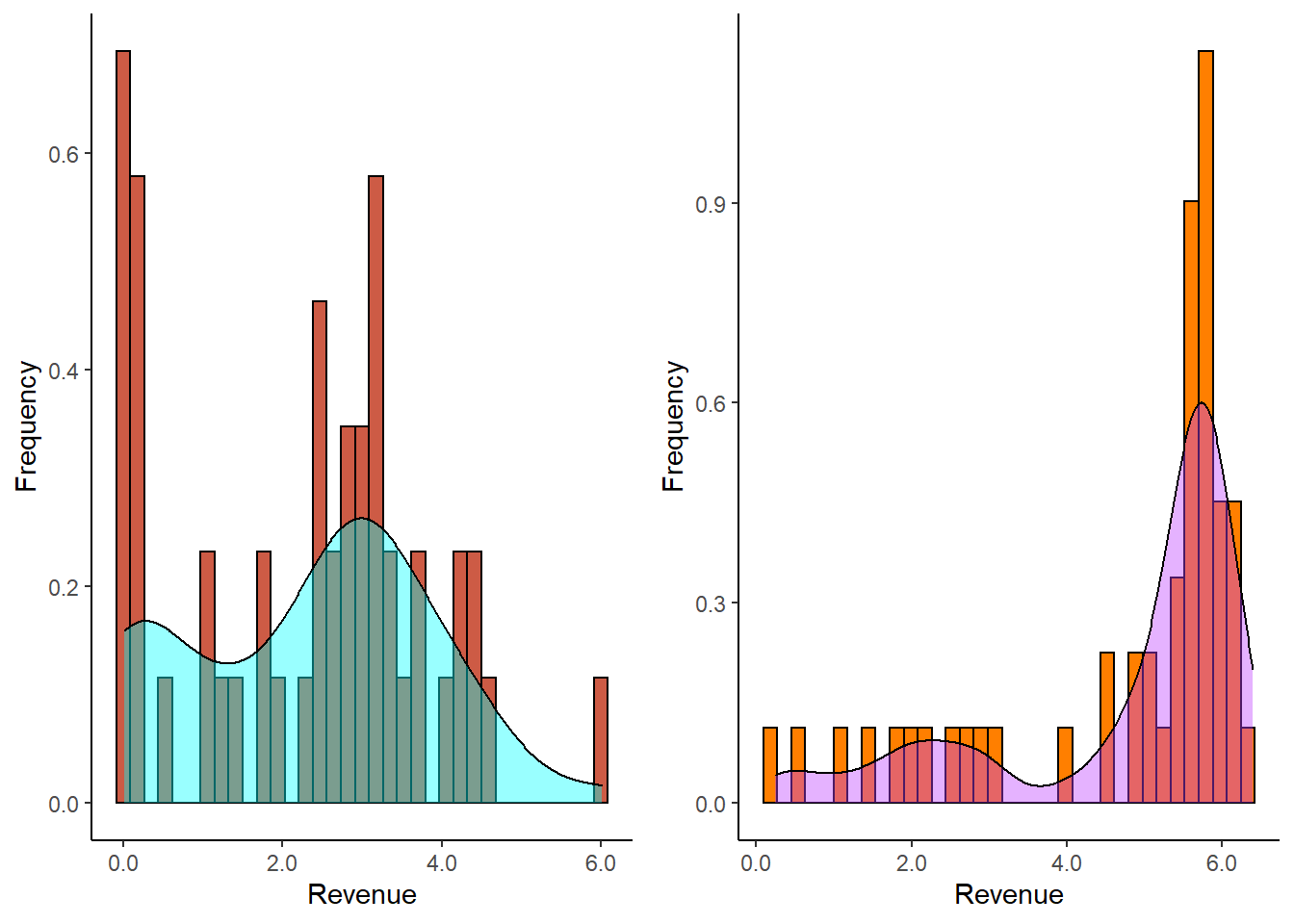
scale\_x\_continuous(labels = comma) +

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, WEEKEND\_DATE) %>% group\_by(MOVIE) %>%

summarise(week\_total = sum(WEEKEND\_PER\_THEATER), week = WEEKEND\_DATE)

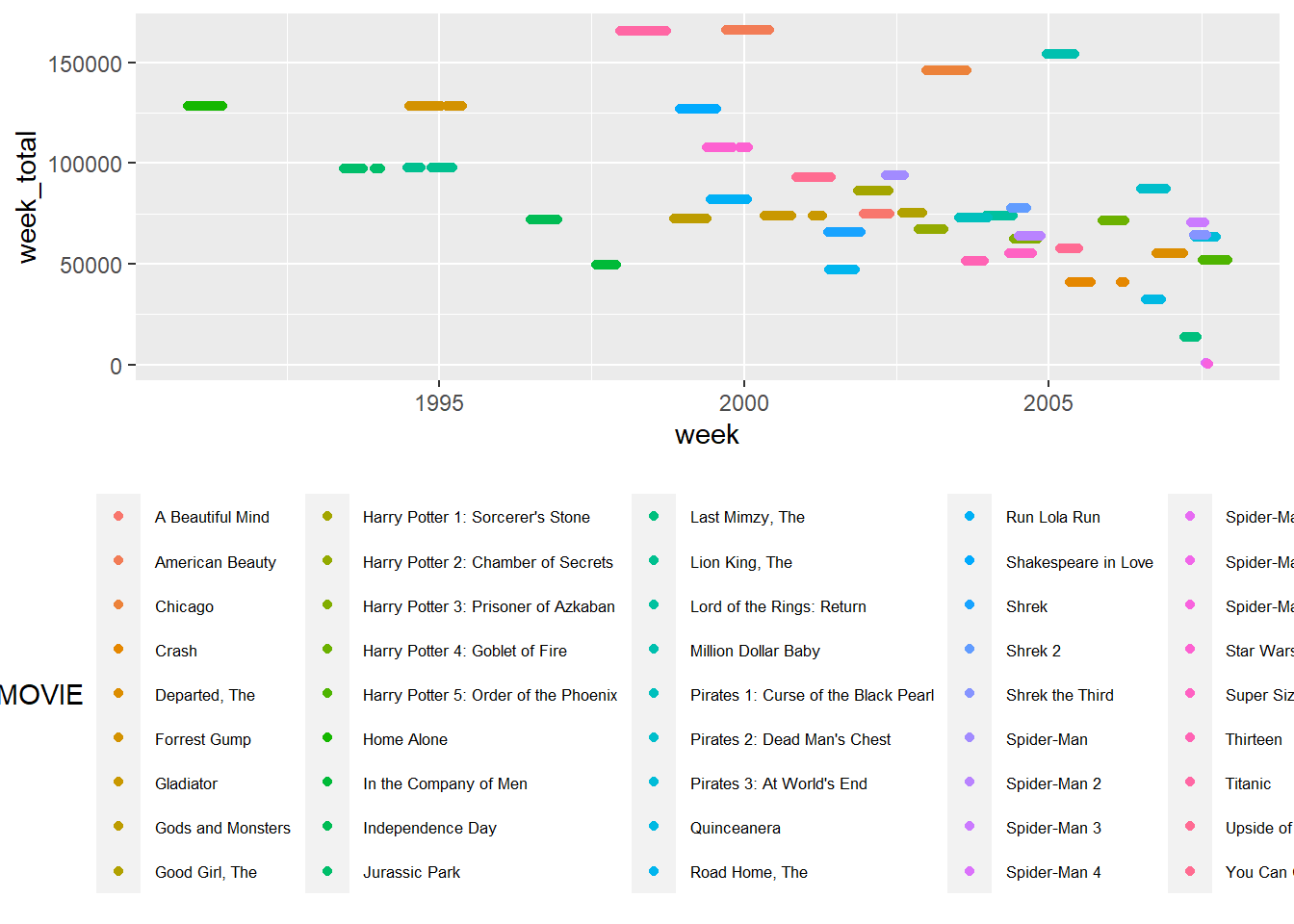
ggplot(mV, aes(x=week, y=week\_total, colour = MOVIE)) +

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")

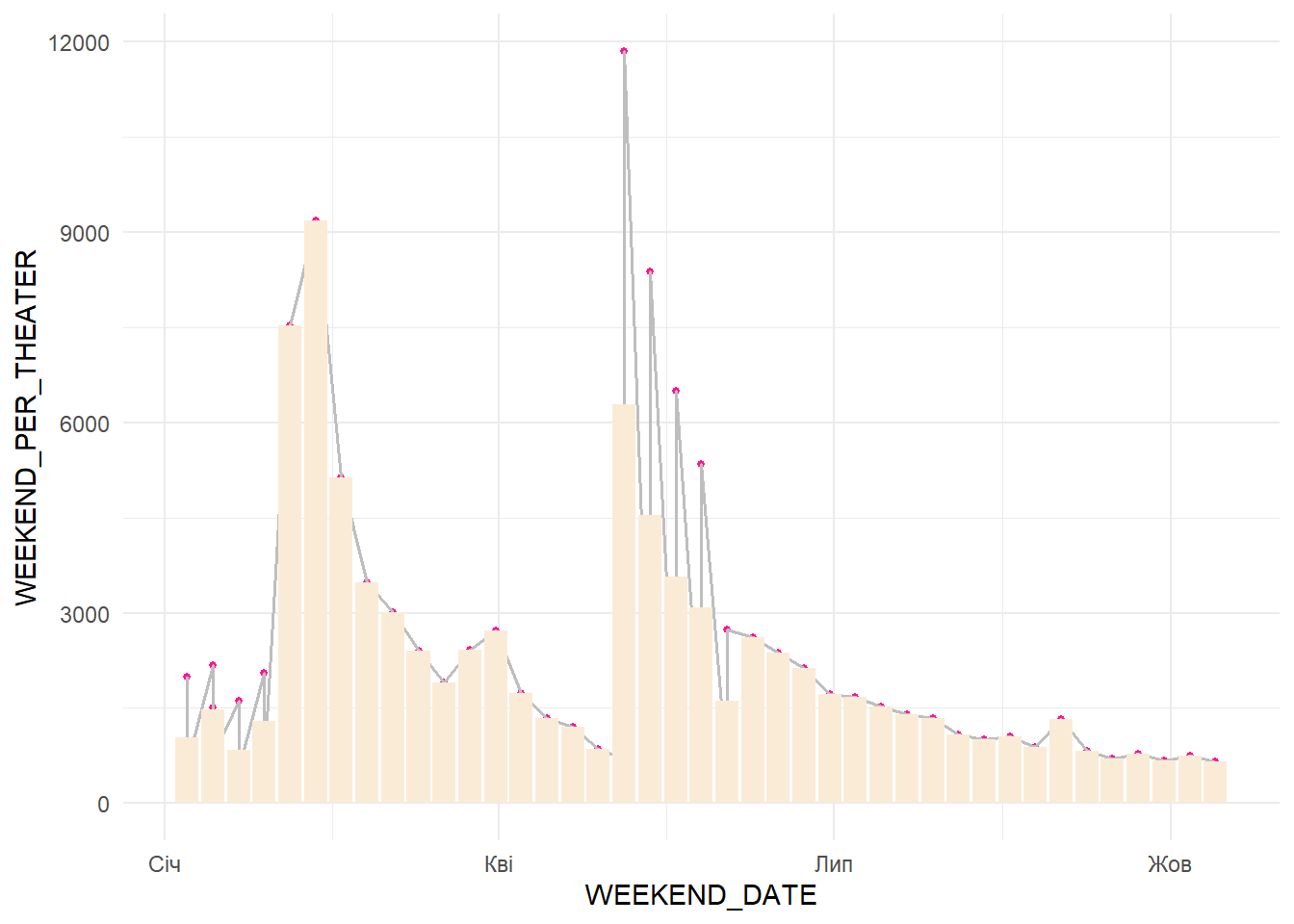
& WEEKEND\_PER\_THEATER >= 200), aes(x = WEEKEND\_DATE, y = WEEKEND\_PER\_THEATER)) +

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()



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)) +

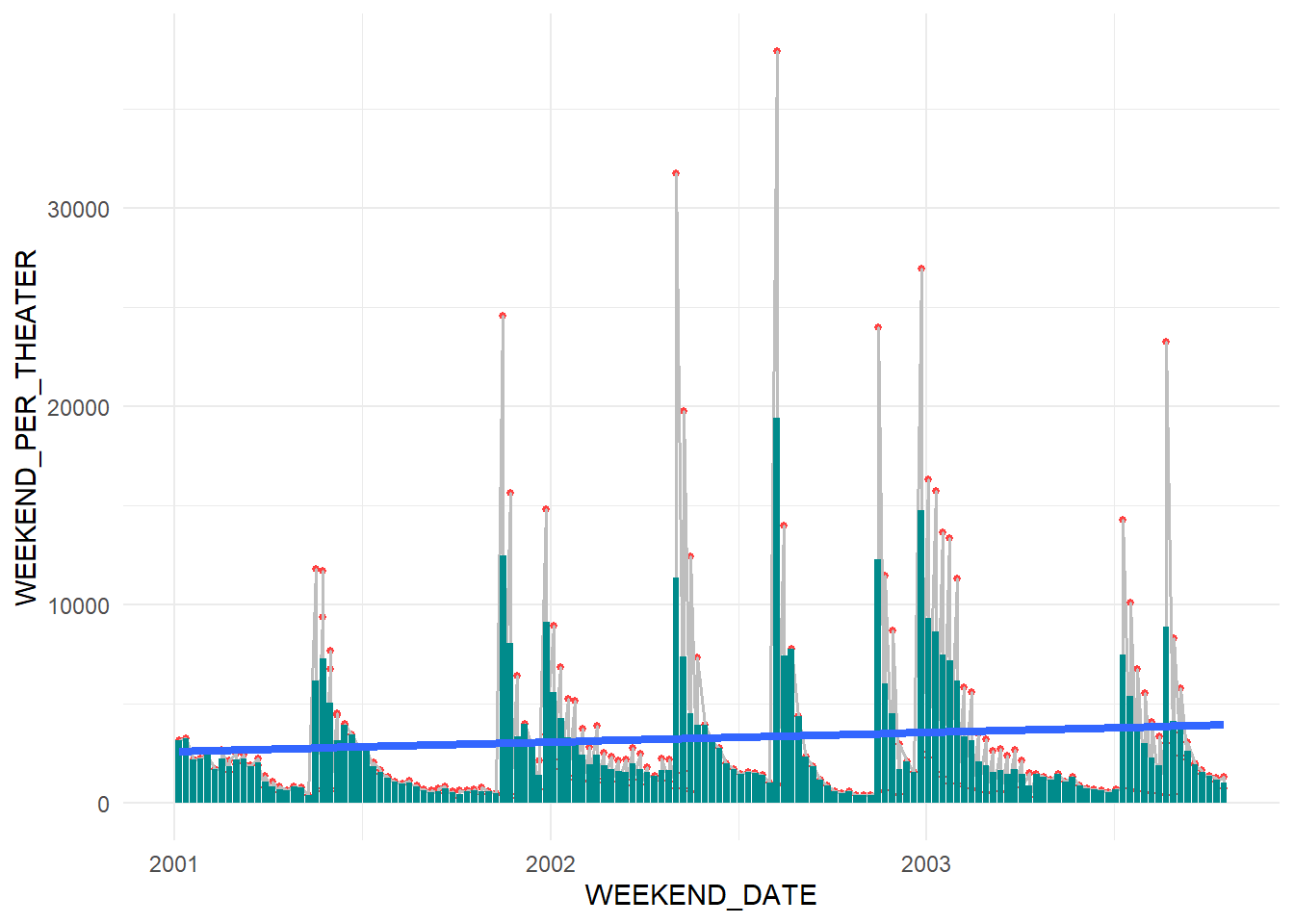
geom\_point(color = "brown1", size = 1) +

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()

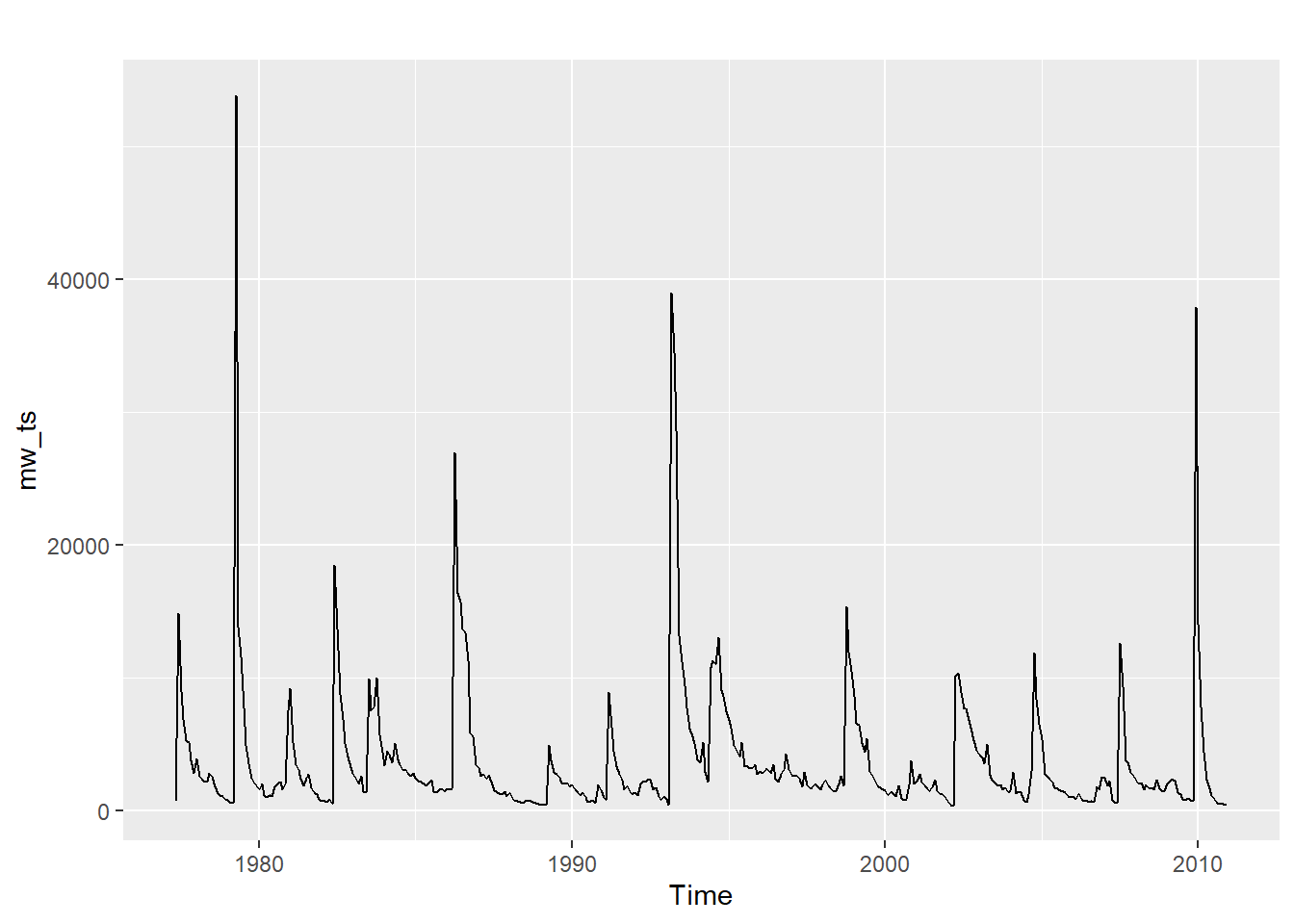


mw\_ts <- ts(movie\_weekend$WEEKEND\_PER\_THEATER, frequency=12, start = c(1977, 05), end = c(2010, 12))

str(mw\_ts)

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

autoplot(mw\_ts)



## 3. Виявлення тенденцій часового ряду

### 3.1. Лінійна модель:

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

##

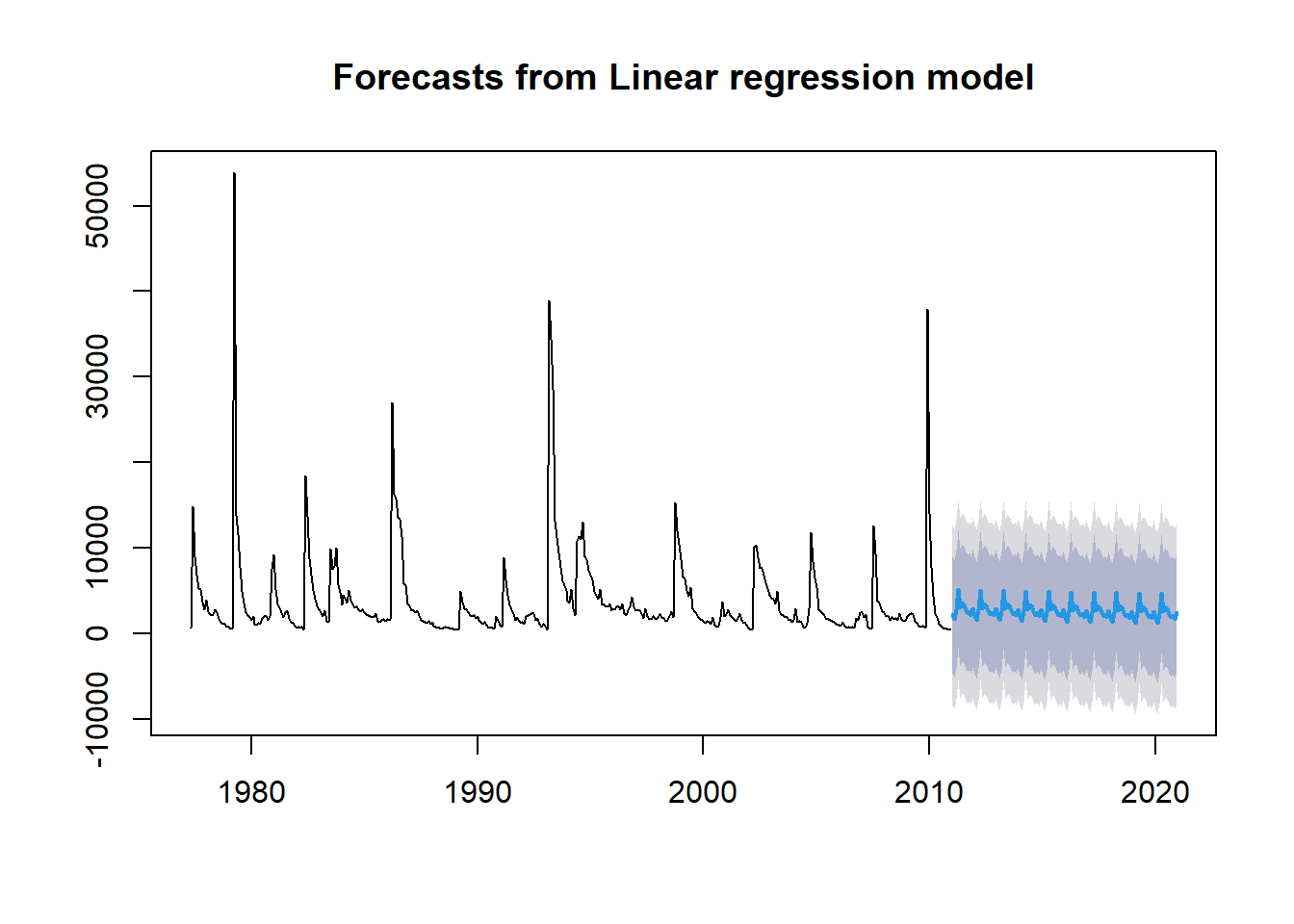
## Residual standard error: 5248 on 391 degrees of freedom

## 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)



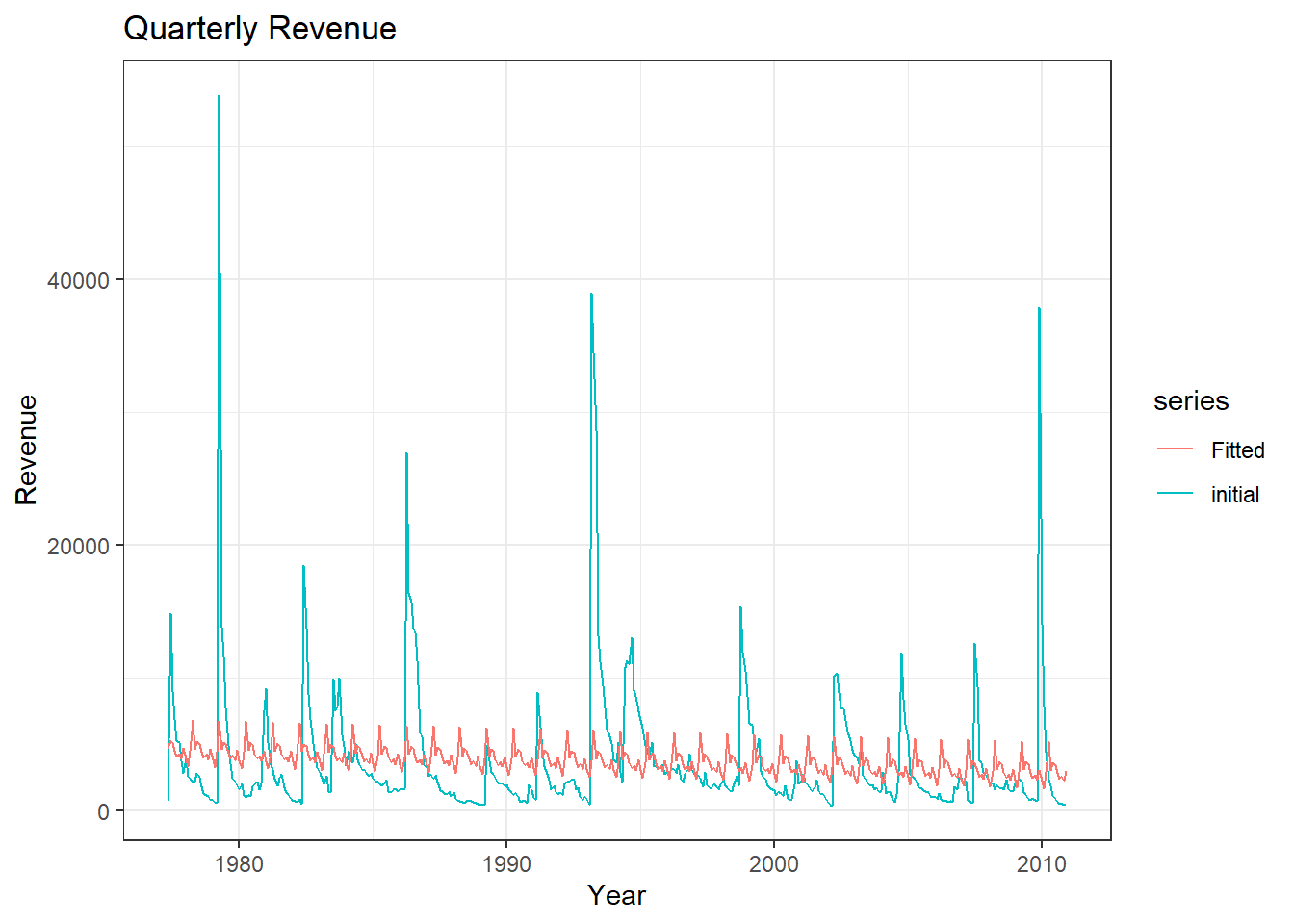
autoplot(mw\_ts, series="initial") +

autolayer(fitted(fit.consMR), series="Fitted") +

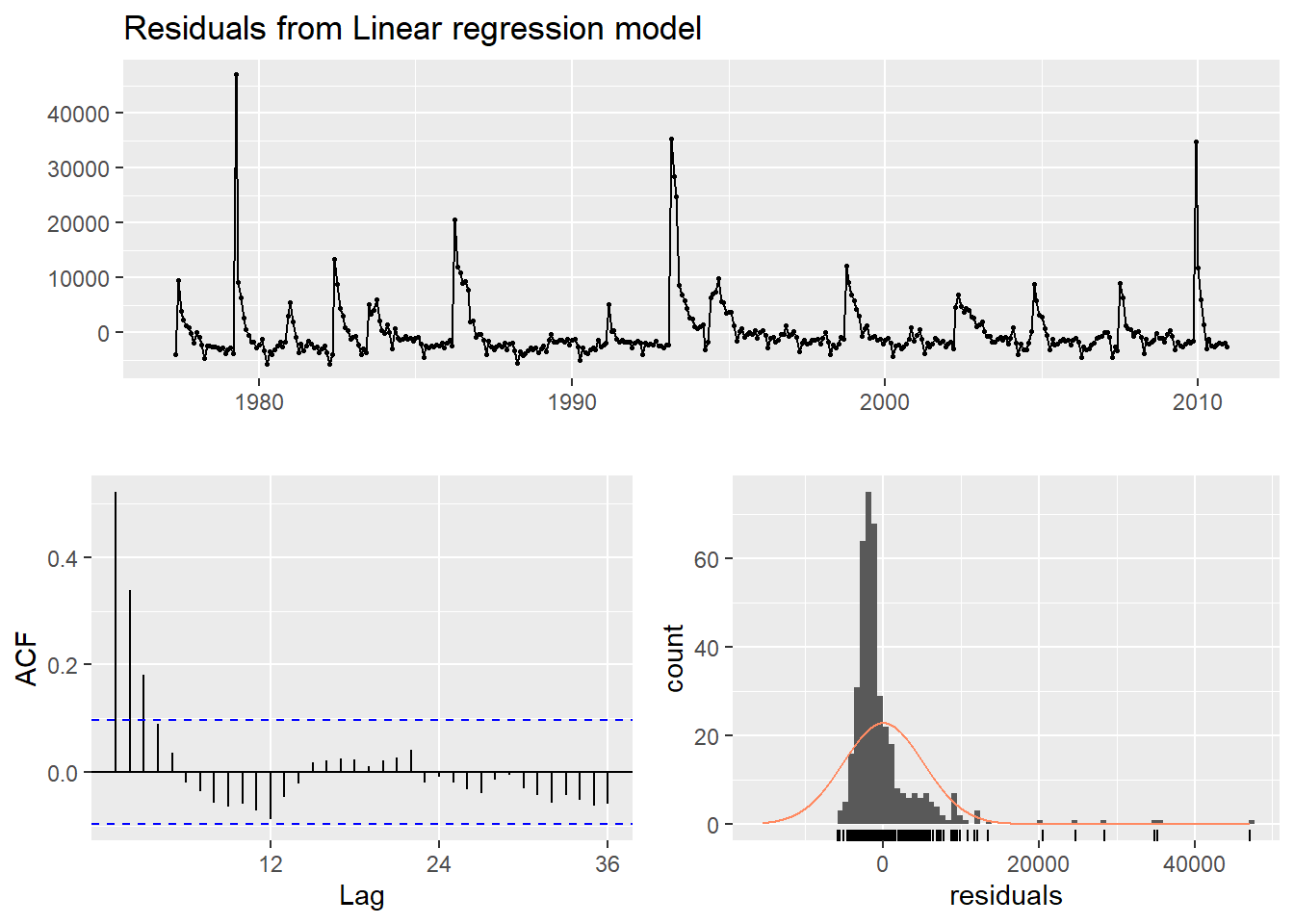
xlab("Year") + ylab("Revenue") +

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

### 3.2. Сплайн, експотенційне, лінійне згладжування

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) +

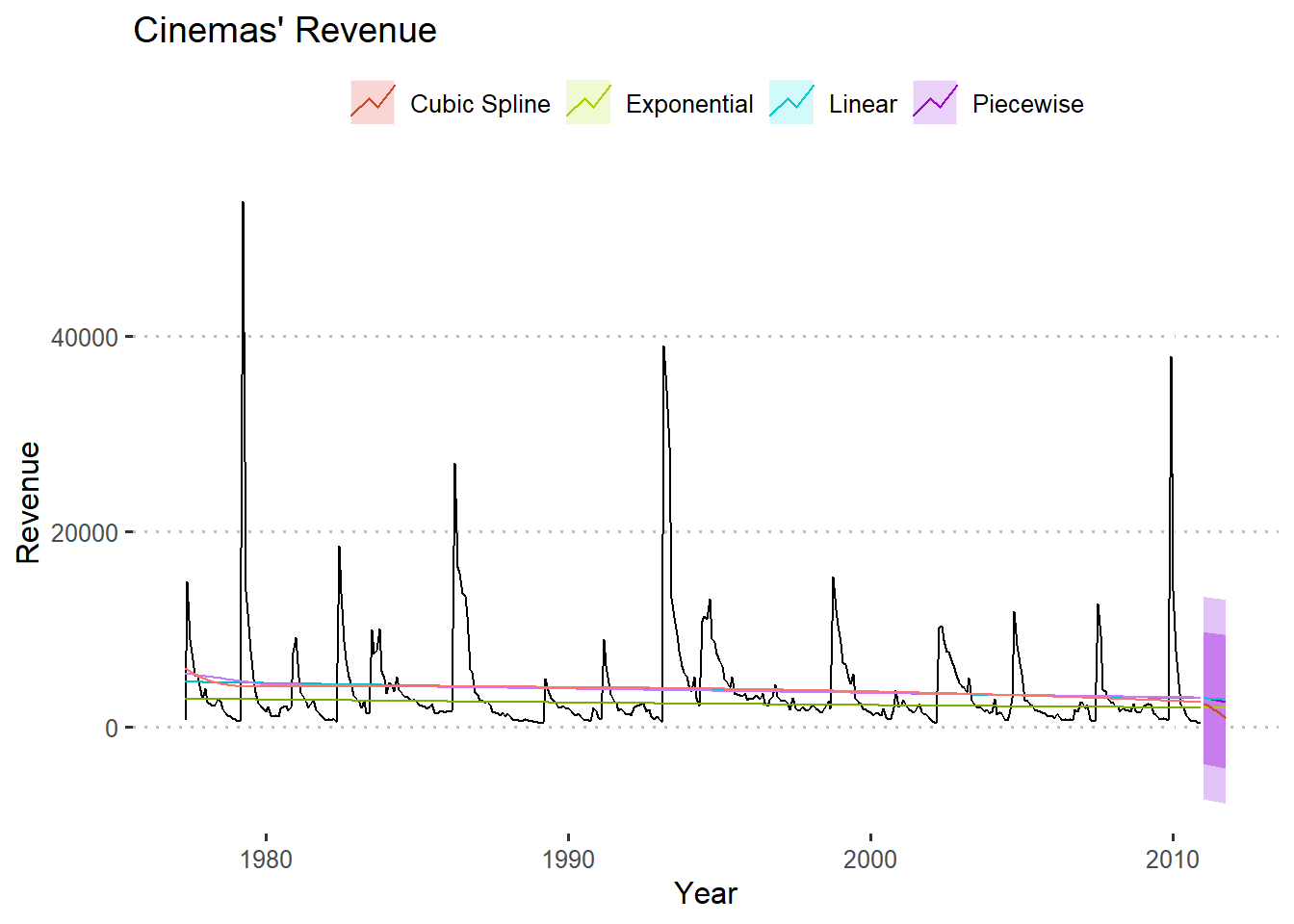
autolayer(fcasts.spl, series="Cubic Spline", PI=FALSE) +

xlab("Year") + ylab("Revenue") +

ggtitle("Cinemas' Revenue") +

guides(colour = guide\_legend(title = " ")) +

theme\_pubclean()



### 3.3. Сплайни та ARIMA-моделі

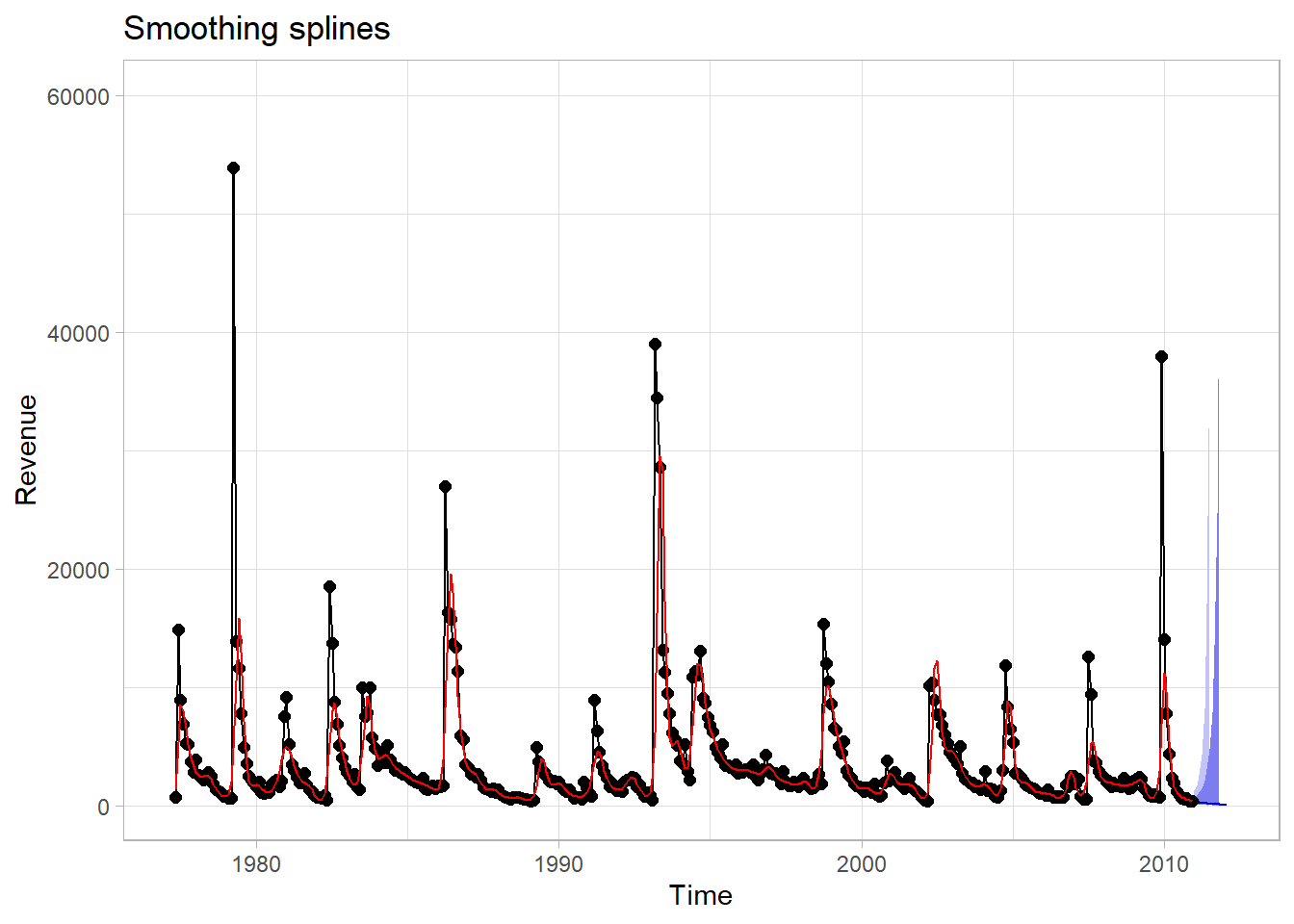
splinef(mw\_ts, h=14, lambda=-0.05, method = "gcv") %>%

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)) +

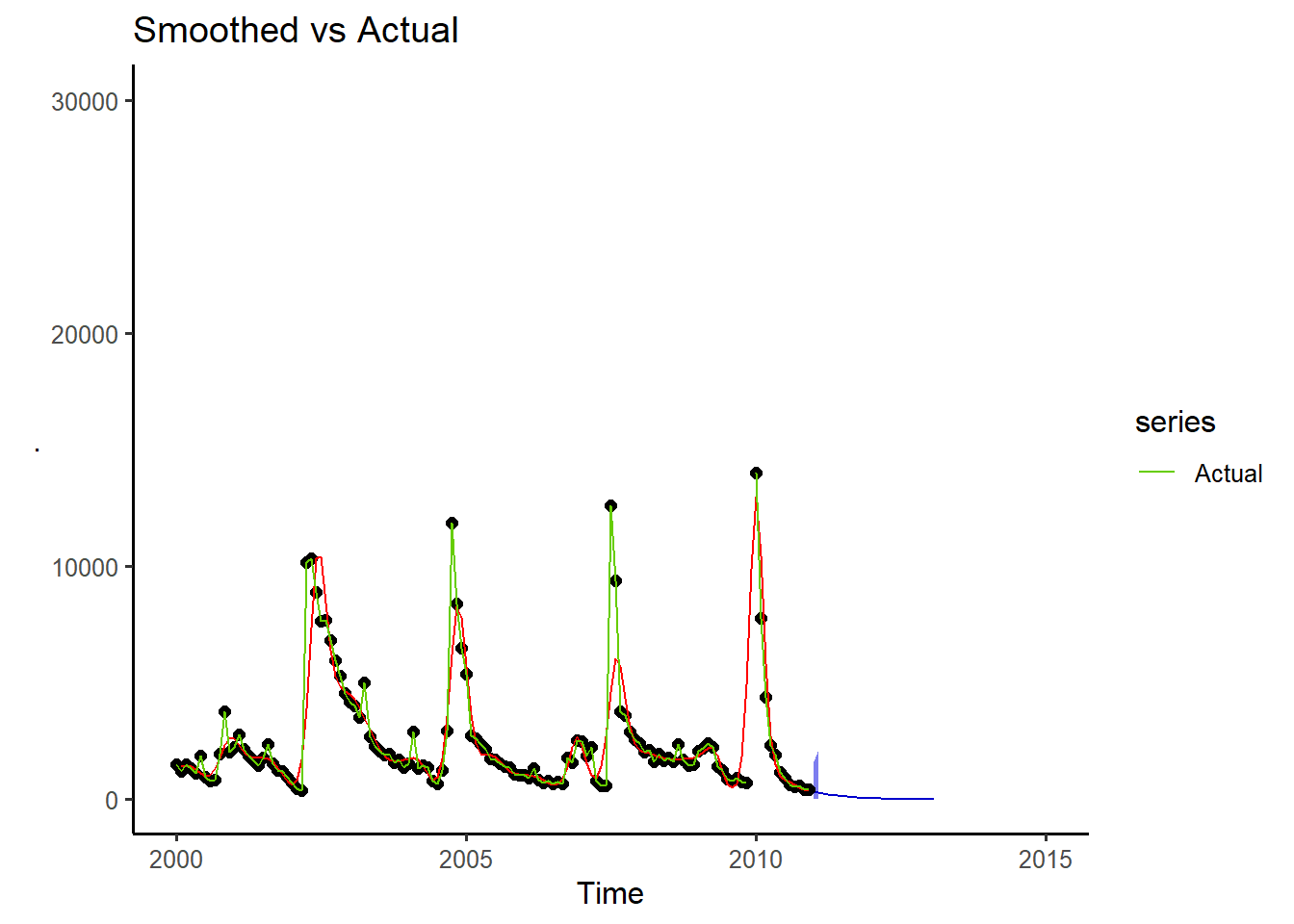
autolayer(mw\_ts) +

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")

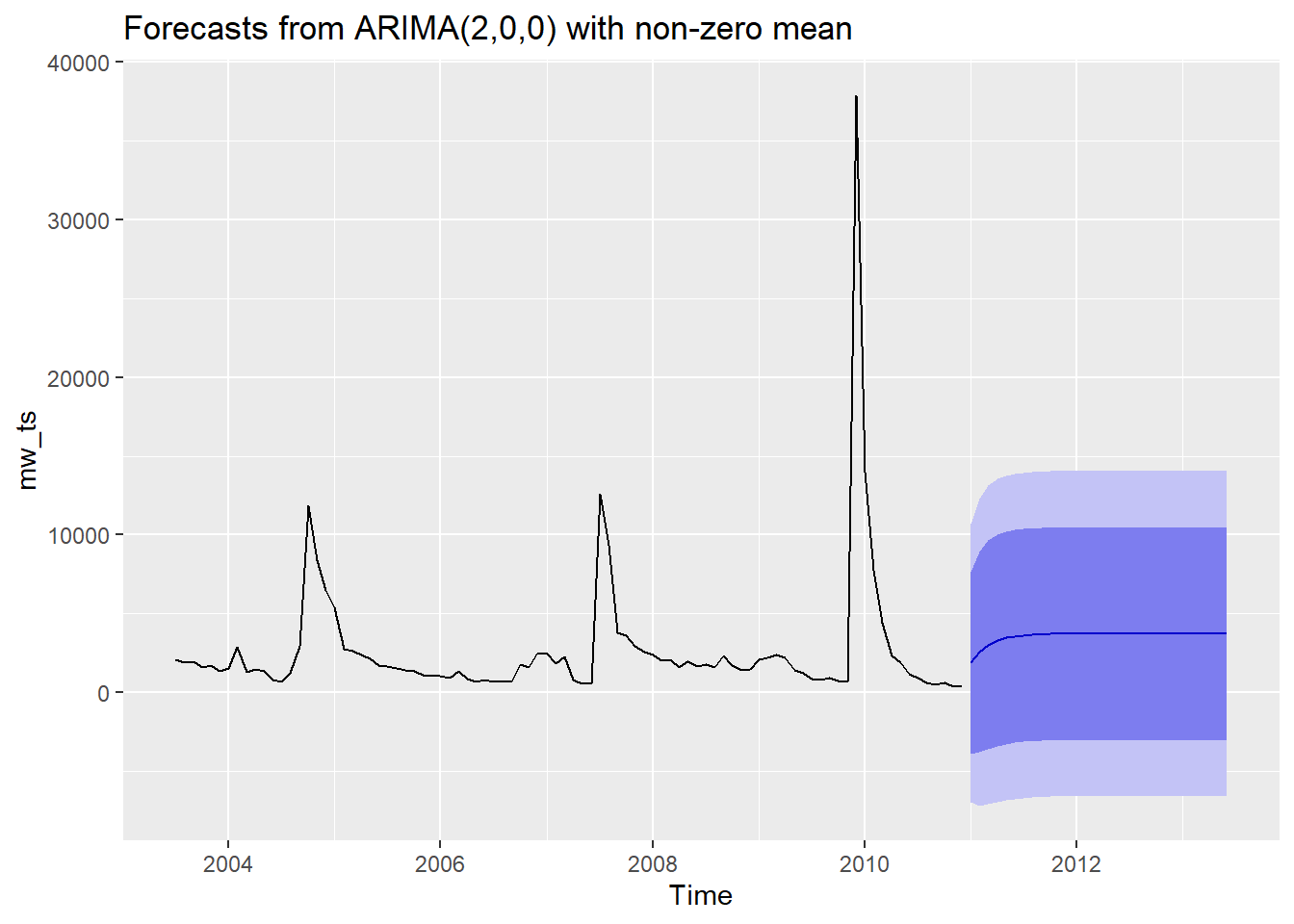
accuracy(spl\_pred)

## 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)



## 4. Кореляційний аналіз часових послідовностей

### 4.1. Random Forest

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 season

## 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",

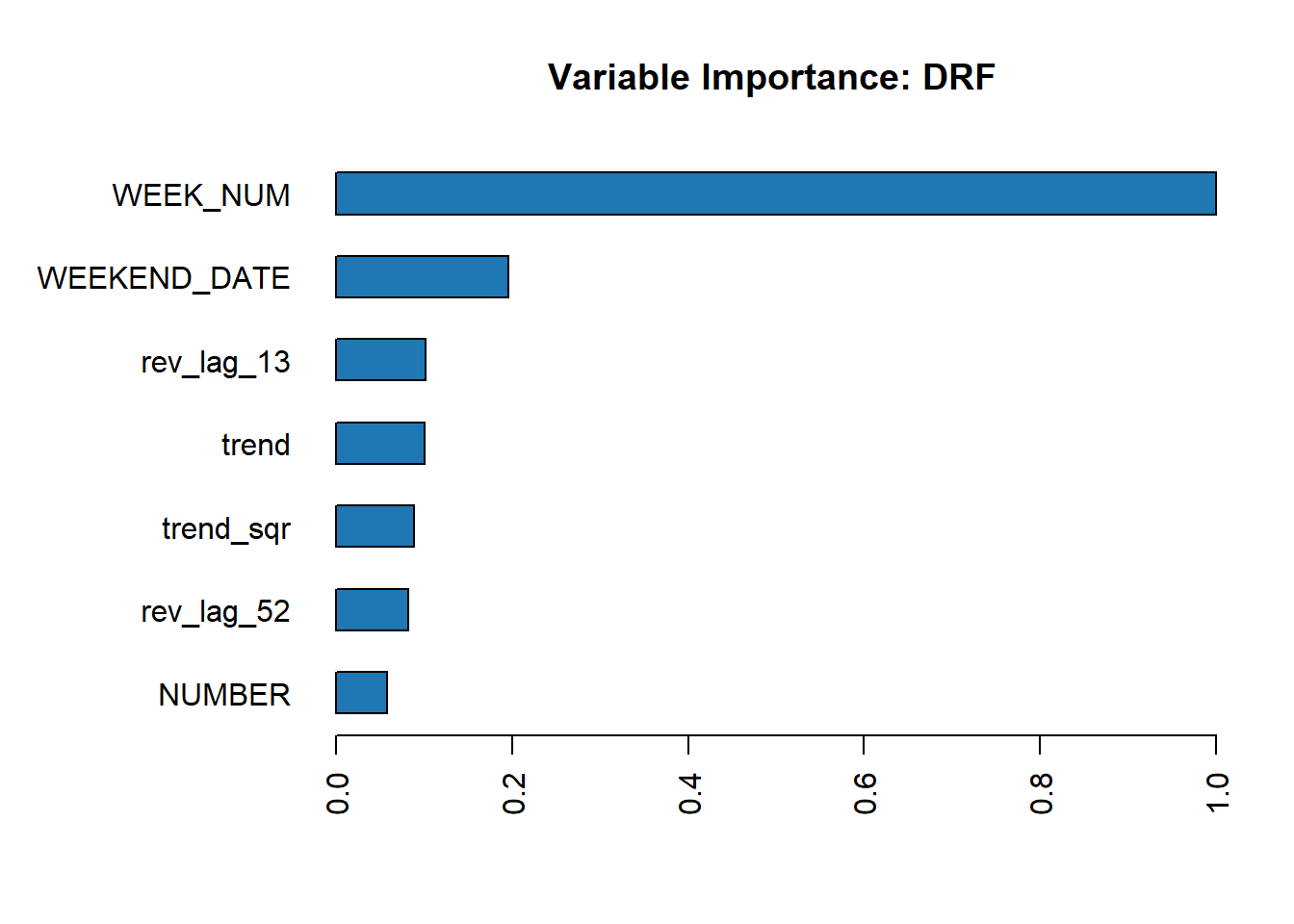
stopping\_rounds = 10,

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

### 4.2. 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,

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

# Висновки:

під час виконання роботи ми вивчили основні методи аналізу статистичних даних, які найчастіше застосовуються в наукових дослідженнях та на практиці, а також сформували практичні навички застосування цих методів за допомогою комп’ютерних програм аналізу даних.