# Chapter 7

## 1. Assignment 1

Consider the two data files Chap\_7\_HW\_File\_2s.csv and Chap\_7\_HW\_File\_3s.csv found in the Files for Homework Assignments folder on BlackBoard. Each file consists of four time series, three predictor time series (P1, P2,P3) and a Data time series which may depend on the three predictors. For each file produce a linear regression model P1, P2,P3, P1 and P2, P1 and P3, P2 and P3, and all three. Use the first 90% of the Data to train the models and final 10% as a test case. Plot each model and the original Data. Analyze the models and let R decide which one is the best model.

library(ggplot2)

library(readr)

library(lubridate)

data\_2s\_df <- read.csv("data/Chap\_7\_HW\_File\_2s.csv")

data\_3s\_df <- read.csv("data/Chap\_7\_HW\_File\_3s.csv")

# Convert to correct date format

data\_2s\_df$Date <- ymd(data\_2s\_df$Date)

data\_3s\_df$Date <- mdy(data\_3s\_df$Date)

# Split the data

split\_data <- function(data) {

total\_rows <- nrow(data)

split\_index <- 1:floor(0.9 \* total\_rows)

train\_df <- data[split\_index, ]

test\_df <- data[-split\_index, ]

list(train\_df = train\_df, test\_df = test\_df)

}

split\_2s\_df <- split\_data(data\_2s\_df)

split\_3s\_df <- split\_data(data\_3s\_df)

# Function to create models

create\_models <- function(train\_df) {

models <- list(

P1 = lm(Data ~ P1, data = train\_df),

P2 = lm(Data ~ P2, data = train\_df),

P3 = lm(Data ~ P3, data = train\_df),

P1\_P2 = lm(Data ~ P1 + P2, data = train\_df),

P1\_P3 = lm(Data ~ P1 + P3, data = train\_df),

P2\_P3 = lm(Data ~ P2 + P3, data = train\_df),

P1\_P2\_P3 = lm(Data ~ P1 + P2 + P3, data = train\_df)

)

return(models)

}

models\_2s <- create\_models(split\_2s\_df$train\_df)

models\_3s <- create\_models(split\_3s\_df$train\_df)

# Evaluate models

evaluate\_models <- function(models, test\_df) {

results\_df <- data.frame(Model = character(),

RMSE = numeric(),

stringsAsFactors = FALSE)

for (name in names(models)) {

model <- models[[name]]

preds <- predict(model, newdata = test\_df)

rmse <- sqrt(mean((test\_df$Data - preds)^2))

results\_df <- rbind(results\_df,

data.frame(Model = name, RMSE = rmse))

}

results\_df <- results\_df[order(results\_df$RMSE), ]

return(results\_df)

}

results\_2s <- evaluate\_models(models\_2s, split\_2s\_df$test\_df)

results\_3s <- evaluate\_models(models\_3s, split\_3s\_df$test\_df)

### Let R decide which one is the best model

stepwise\_2s <- stepAIC(lm(Data ~ P1 + P2 + P3, data = split\_2s\_df$train\_df),

direction = "both")

stepwise\_3s <- stepAIC(lm(Data ~ P1 + P2 + P3, data = split\_3s\_df$train\_df),

direction = "both")

> summary(stepwise\_2s)

Call:

lm(formula = Data ~ P1 + P2 + P3, data = split\_2s\_df$train\_df)

Residuals:

Min 1Q Median 3Q Max

-1.01377 -0.49274 -0.01619 0.49471 1.03449

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 0.055097 0.063313 0.87 0.384

P1 1.996318 0.005405 369.37 <2e-16 \*\*\*

P2 2.995278 0.005546 540.13 <2e-16 \*\*\*

P3 -1.505126 0.005574 -270.05 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5769 on 1310 degrees of freedom

Multiple R-squared: 0.9974, Adjusted R-squared: 0.9974

F-statistic: 1.685e+05 on 3 and 1310 DF, p-value: < 2.2e-16

> summary(stepwise\_3s)

Call:

lm(formula = Data ~ P1 + P2, data = split\_3s\_df$train\_df)

Residuals:

Min 1Q Median 3Q Max

-5.0968 -2.9558 -0.0017 2.9568 5.0829

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 5.00630 0.04628 108.18 <2e-16 \*\*\*

P1 1.01740 0.04319 23.55 <2e-16 \*\*\*

P2 1.01873 0.02179 46.76 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.139 on 4599 degrees of freedom

Multiple R-squared: 0.3727, Adjusted R-squared: 0.3724

F-statistic: 1366 on 2 and 4599 DF, p-value: < 2.2e-16

Conclusions:

* stepwise\_2s is clearly a much better model since ~99.7% of variance explained (R2) and all predictors (P1, P2, P3) are highly significant.
* For stepwise\_3s, the best model is the one R selected (P1, P2), but even that model is weak. The model explains only ~37% of the variance (R2)

### Plots

plot\_best\_model <- function(best\_model, test\_df) {

preds <- predict(best\_model, newdata = test\_df)

plot\_df <- data.frame(Actual = test\_df$Data, Predicted = preds)

ggplot(plot\_df, aes(x = Actual, y = Predicted)) +

geom\_line(size=1) +

geom\_abline(intercept = 0, slope = 1, linetype = "dashed", color = "blue") +

labs(x = "Test data",

y = "Predicted") +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

}

plot\_best\_model(stepwise\_2s, split\_2s\_df$test\_df)

plot\_best\_model(stepwise\_3s, split\_3s\_df$test\_df)

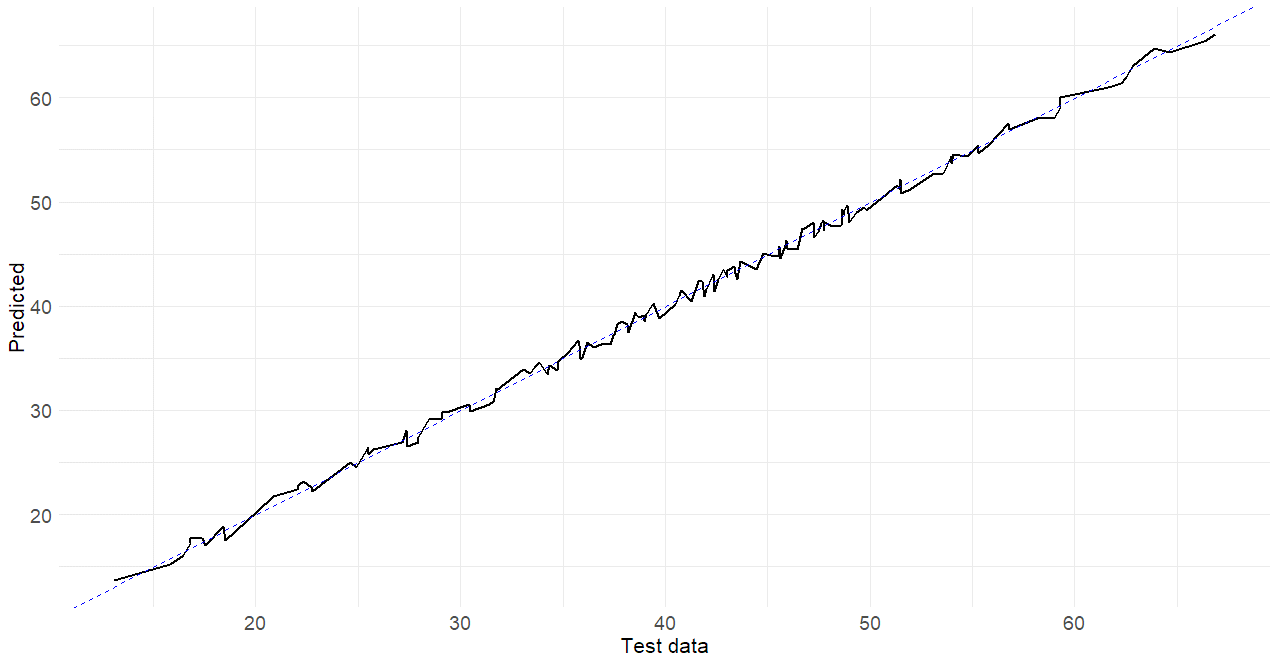


Figure 1: Test data vs predicted (2s file)

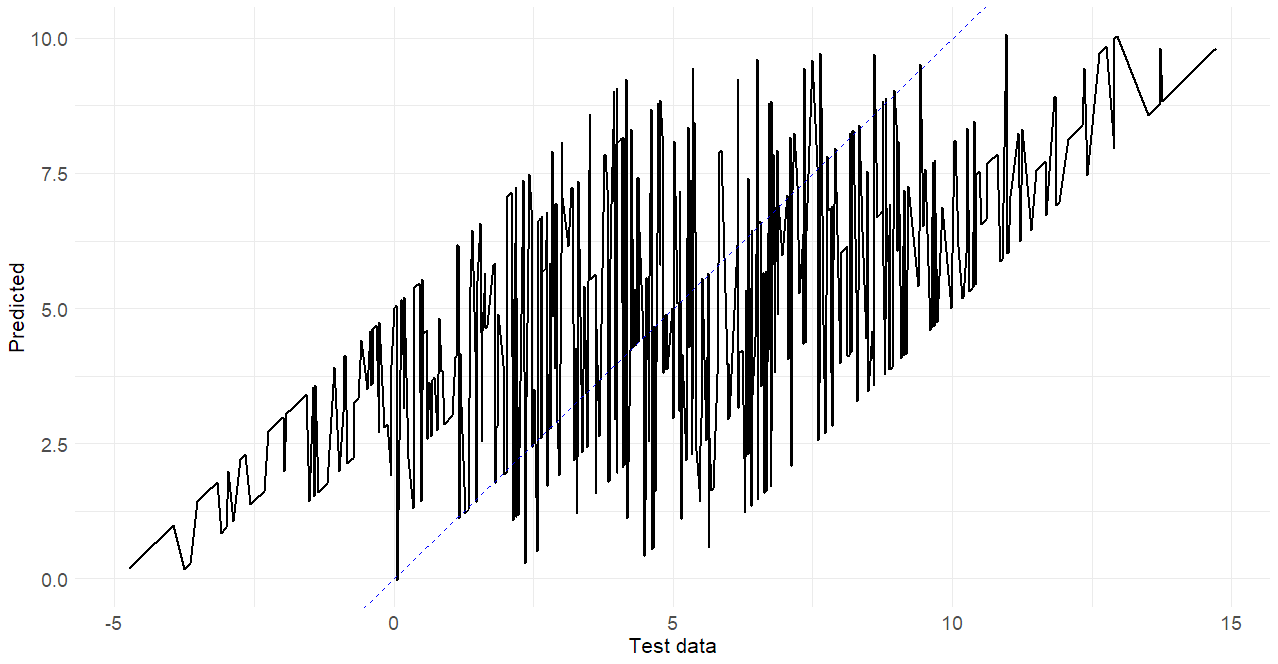


Figure 2: Test data vs predicted (3s file)

Conclusions: stepwise\_2s is clearly a much better model with close prediction with the test data

## 2. Assignment 2

The annual population of Afghanistan is available in the global\_economy data set.

### a. Plot the data and comment on its features.

afghanistan\_population <- global\_economy %>%

filter(Country == "Afghanistan")

afghanistan\_population %>%

tsibble(index = Year)%>%

ggplot(aes(x = Year, y = Population)) +

geom\_line(size=1) +

labs(x = "Year",

y = "Population") +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

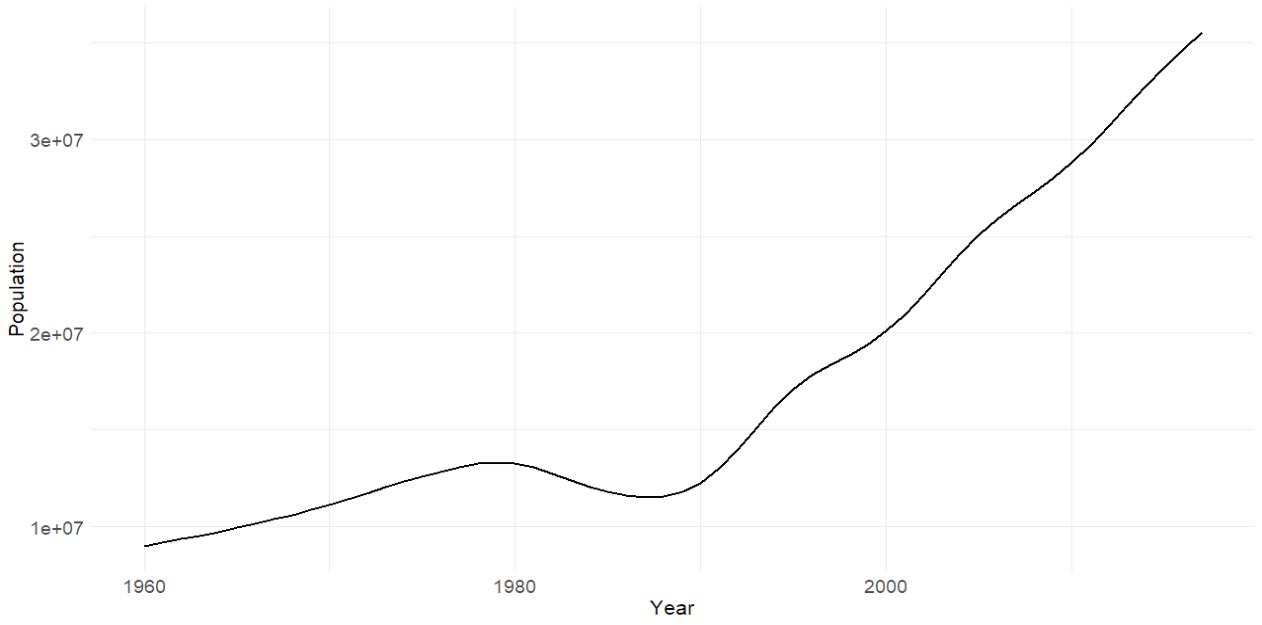


Figure 3: Time Series: Afghanistan Population vs Year

The population grew slowly from 1960 to 1980, then began to decline until 1990, after which it increased rapidly.

## b. Fit a linear trend model and compare this to a piecewise linear trend model with knots at 1980 and 1989.

### Linear trend model

linear\_trend\_model <- afghanistan\_population %>%

model(Linear = TSLM(Population ~ trend()))

linear\_trend\_model %>% report()

Series: Population

Model: TSLM

Residuals:

Min 1Q Median 3Q Max

-5794518 -2582559 744761 2259222 6036280

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4798904 848259 5.657 5.45e-07 \*\*\*

trend() 425774 25008 17.025 < 2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3188000 on 56 degrees of freedom

Multiple R-squared: 0.8381, Adjusted R-squared: 0.8352

F-statistic: 289.9 on 1 and 56 DF, p-value: < 2.22e-16

### Piecewise linear trend model

piecewise\_linear\_trend\_model <- afghanistan\_population %>%

model(Piecewise\_Linear = TSLM(Population ~ trend(knots = c(1980, 1989))))

piecewise\_linear\_trend\_model %>% report()

Series: Population

Model: TSLM

Residuals:

Min 1Q Median 3Q Max

-577590 -174198 -16784 187226 679947

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8697573 131122 66.33 <2e-16 \*\*\*

trend(knots = c(1980, 1989))trend 224372 9623 23.32 <2e-16 \*\*\*

trend(knots = c(1980, 1989))trend\_21 -456804 24498 -18.65 <2e-16 \*\*\*

trend(knots = c(1980, 1989))trend\_30 1082782 21418 50.55 <2e-16 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 300900 on 54 degrees of freedom

Multiple R-squared: 0.9986, Adjusted R-squared: 0.9985

F-statistic: 1.293e+04 on 3 and 54 DF, p-value: < 2.22e-16

### Compare the models visually

fit <- afghanistan\_population %>%

model(

Linear = TSLM(Population ~ trend()),

Piecewise\_Linear = TSLM(Population ~ trend(knots = c(1980, 1989)))

)

augment(fit) %>%

autoplot(.fitted) +

geom\_line(aes(y = Population), colour = "black", size=1) +

labs(x = "Year",

y = "Population") +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)



Figure 4: Linear trend model vs Piecewise linear trend model

Based on the graph and from the reports, the piecewise linear model explains 99.86% of the variance in the data, much higher than the 83.81% explained by the simple linear model. Its residual error is also about 90% lower, indicating much better accuracy.

### c. Generate forecasts from these two models for the five years after the end of the data, and comment on the results.

next\_5\_years\_forecast <- fit %>% forecast(h = "5 years")

autoplot(next\_5\_years\_forecast) +

autolayer(afghanistan\_population, Population) +

theme\_minimal() +

theme(

axis.title.x = element\_text(size = 16, hjust = 0.5),

axis.title.y = element\_text(size = 16),

axis.text.x = element\_text(size = 14),

axis.text.y = element\_text(size = 14),

legend.title = element\_text(size = 16),

legend.text = element\_text(size = 14)

)

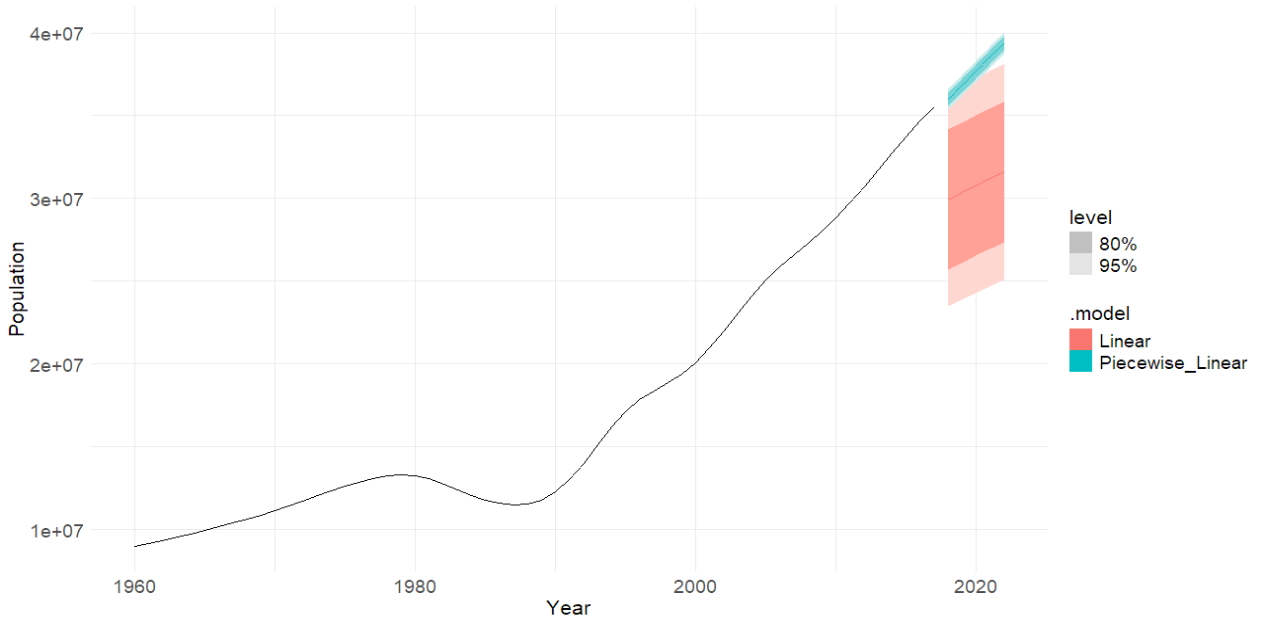


Figure 5: Forecasts from two models

The linear model appears inadequate, as it produces excessively wide prediction intervals and systematically underestimates future values.

In contrast, the piecewise linear model provides a better fit to the data. However, its prediction intervals are likely too narrow, reflecting overconfidence in the model's assumptions.