Project 2

Jaymeen Gandhi

12/8/2019

R Markdown

Predictive Analysis

**require**(tidyverse)

## Loading required package: tidyverse

## -- Attaching packages -------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.2

## v tibble 2.1.3 v dplyr 0.8.3

## v tidyr 1.0.0 v stringr 1.4.0

## v readr 1.3.1 v forcats 0.4.0

## -- Conflicts ----------------------------------------------------------------------------------- tidyverse\_conflicts() --

## x dplyr::filter() masks stats::filter()

## x dplyr::lag() masks stats::lag()

**library**(dplyr)

**library**(lubridate)

##

## Attaching package: 'lubridate'

## The following object is masked from 'package:base':

##

## date

**library**(ggplot2)

**library**(forecast)

## Registered S3 method overwritten by 'xts':

## method from

## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':

## method from

## as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':

## method from

## fitted.fracdiff fracdiff

## residuals.fracdiff fracdiff

bikeds = read\_csv("C:/Users/jayme/Desktop/bikeds.csv")

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:

## cols(

## X1 = col\_double(),

## `Trip Duration` = col\_double(),

## Date = col\_date(format = ""),

## month = col\_character(),

## `Start Time` = col\_time(format = ""),

## `Stop Time` = col\_time(format = ""),

## `Start Station ID` = col\_double(),

## `Start Station Name` = col\_character(),

## `Start Station Latitude` = col\_double(),

## `Start Station Longitude` = col\_double(),

## `End Station ID` = col\_double(),

## `End Station Name` = col\_character(),

## `End Station Latitude` = col\_double(),

## `End Station Longitude` = col\_double(),

## `Bike ID` = col\_double(),

## `User Type` = col\_character(),

## `Birth Year` = col\_double(),

## Gender = col\_double(),

## Trip\_Duration\_in\_min = col\_double()

## )

bike\_ds = bikeds %>% select(Date,Trip\_Duration\_in\_min)

colnames(bike\_ds)<-c("date","duration")

bike\_ds = bike\_ds %>% group\_by(date)%>%summarize(md=mean(duration))

bike\_ds

## # A tibble: 550 x 2

## date md

## <date> <dbl>

## 1 2015-09-21 24.5

## 2 2015-09-22 20.3

## 3 2015-09-23 18.1

## 4 2015-09-24 23.0

## 5 2015-09-25 13.7

## 6 2015-09-26 366.

## 7 2015-09-27 32.7

## 8 2015-09-28 15.2

## 9 2015-09-29 14.3

## 10 2015-09-30 14.3

## # ... with 540 more rows

str(bike\_ds)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 550 obs. of 2 variables:

## $ date: Date, format: "2015-09-21" "2015-09-22" ...

## $ md : num 24.5 20.3 18.1 23 13.7 ...

## - attr(\*, "spec")=

## .. cols(

## .. X1 = col\_double(),

## .. `Trip Duration` = col\_double(),

## .. Date = col\_date(format = ""),

## .. month = col\_character(),

## .. `Start Time` = col\_time(format = ""),

## .. `Stop Time` = col\_time(format = ""),

## .. `Start Station ID` = col\_double(),

## .. `Start Station Name` = col\_character(),

## .. `Start Station Latitude` = col\_double(),

## .. `Start Station Longitude` = col\_double(),

## .. `End Station ID` = col\_double(),

## .. `End Station Name` = col\_character(),

## .. `End Station Latitude` = col\_double(),

## .. `End Station Longitude` = col\_double(),

## .. `Bike ID` = col\_double(),

## .. `User Type` = col\_character(),

## .. `Birth Year` = col\_double(),

## .. Gender = col\_double(),

## .. Trip\_Duration\_in\_min = col\_double()

## .. )

bike\_ds$date <- strptime(bike\_ds$date, "%Y-%m-%d" )

bike\_ds$date <- as.POSIXct(bike\_ds$date)

bike\_ds$duration <- as.numeric(unlist(bike\_ds$md))

bike\_ds

## # A tibble: 550 x 3

## date md duration

## <dttm> <dbl> <dbl>

## 1 2015-09-21 00:00:00 24.5 24.5

## 2 2015-09-22 00:00:00 20.3 20.3

## 3 2015-09-23 00:00:00 18.1 18.1

## 4 2015-09-24 00:00:00 23.0 23.0

## 5 2015-09-25 00:00:00 13.7 13.7

## 6 2015-09-26 00:00:00 366. 366.

## 7 2015-09-27 00:00:00 32.7 32.7

## 8 2015-09-28 00:00:00 15.2 15.2

## 9 2015-09-29 00:00:00 14.3 14.3

## 10 2015-09-30 00:00:00 14.3 14.3

## # ... with 540 more rows

*#To get the month and the year*

bike\_ds <- mutate(bike\_ds, MonthYear = paste(year(date),formatC(month(date), width = 2, flag = "0")))

*#To get the week of the year*

bike\_ds <- mutate(bike\_ds, Week = week(date))

*#To seperate the year from the date*

bike\_ds <- mutate(bike\_ds, Year = year(date))

bike\_ds$Year <- as.factor(bike\_ds$Year)

str(bike\_ds)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 550 obs. of 6 variables:

## $ date : POSIXct, format: "2015-09-21" "2015-09-22" ...

## $ md : num 24.5 20.3 18.1 23 13.7 ...

## $ duration : num 24.5 20.3 18.1 23 13.7 ...

## $ MonthYear: chr "2015 09" "2015 09" "2015 09" "2015 09" ...

## $ Week : num 38 38 38 39 39 39 39 39 39 39 ...

## $ Year : Factor w/ 3 levels "2015","2016",..: 1 1 1 1 1 1 1 1 1 1 ...

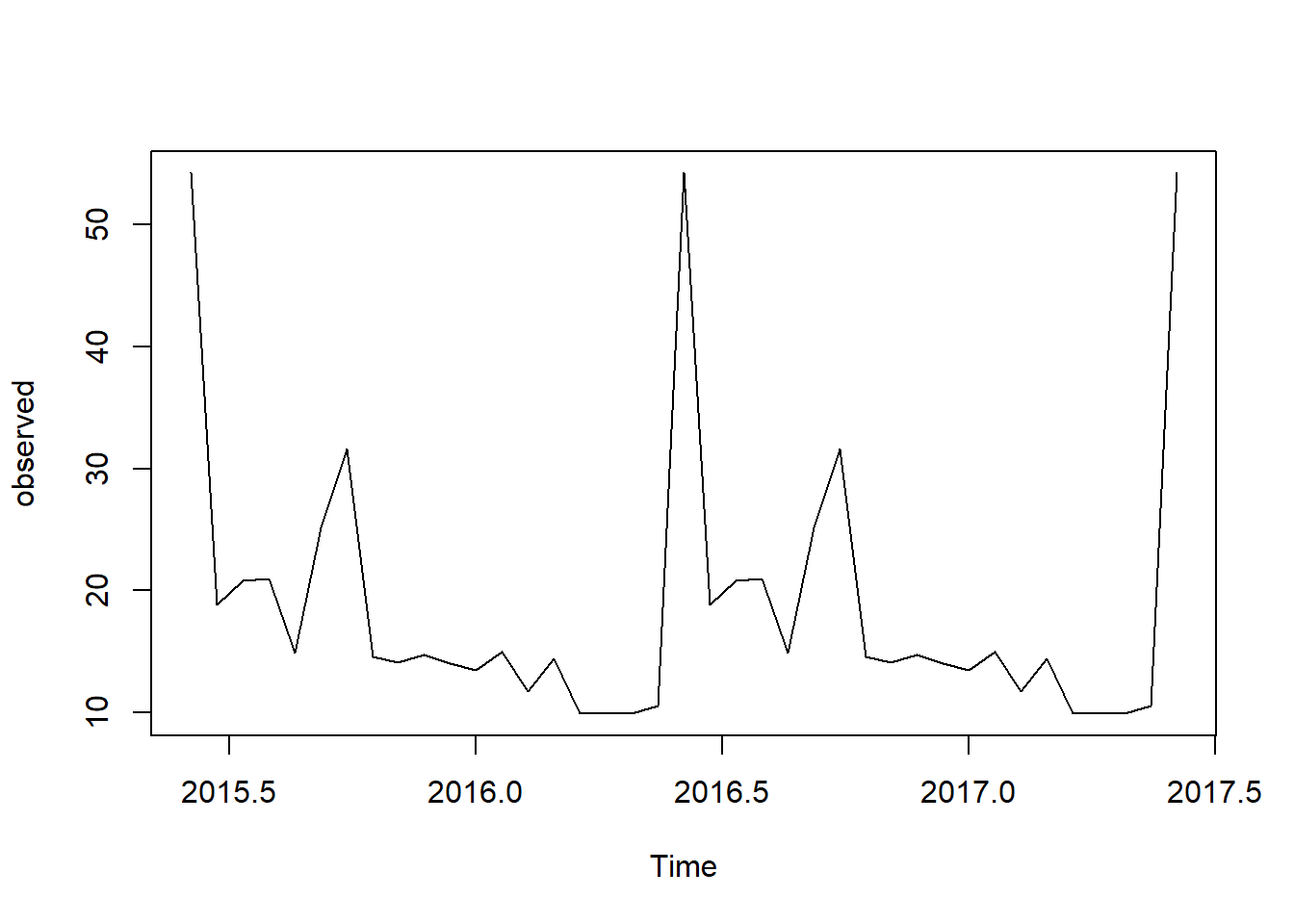
*#duration on monthly basis*

bike\_ds\_month <- aggregate(bike\_ds$duration, by = list(bike\_ds$MonthYear), FUN = **function**(x) mean(x, na.rm=T))

*#plotting the observed flow of the duration throughout the period*

observed <- ts(bike\_ds\_month$x, frequency=19, start = c(2015, 9), end = c(2017, 9))

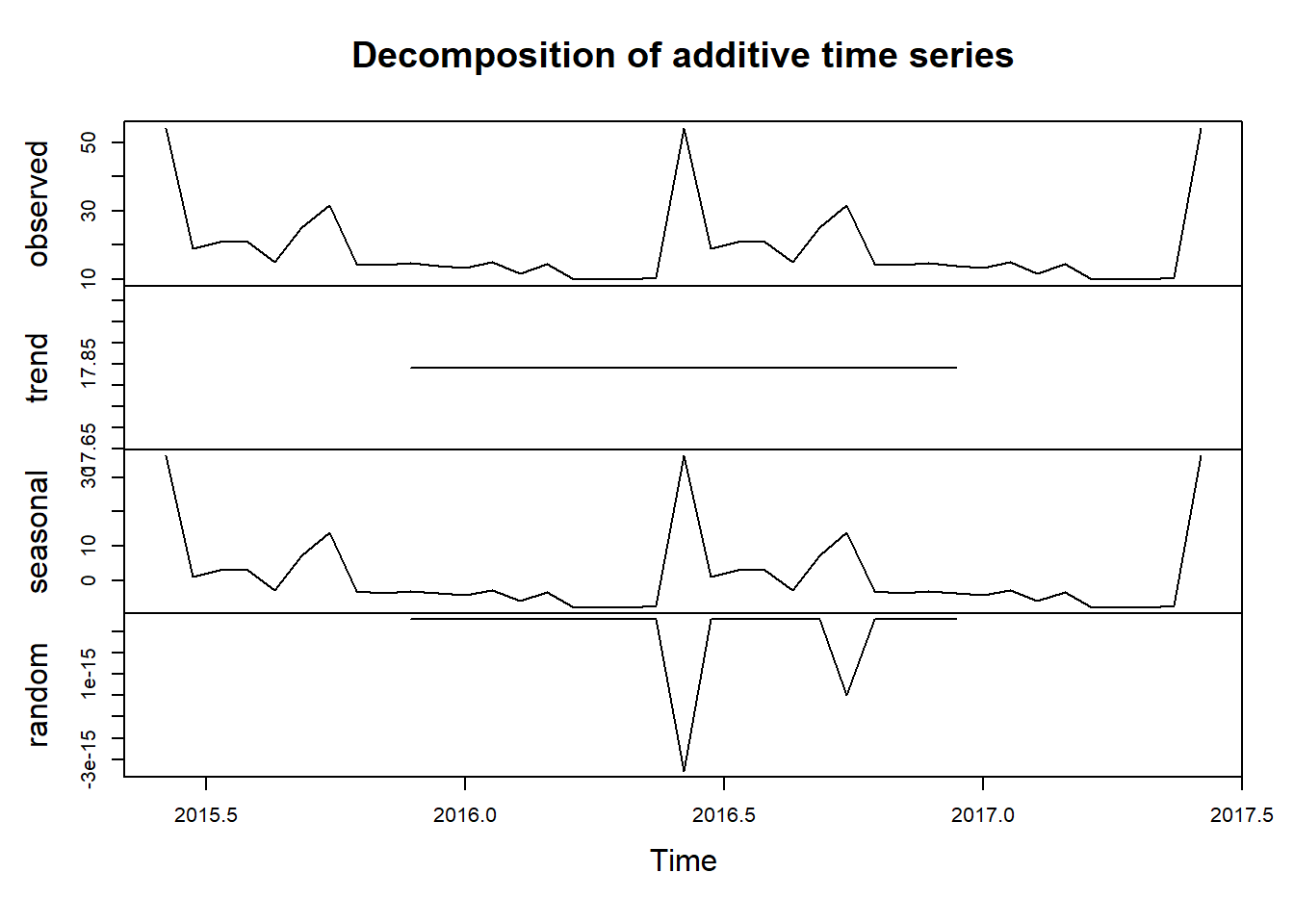
plot(observed)



*#decomposing the dataset into trend, seasonality and random(error)*

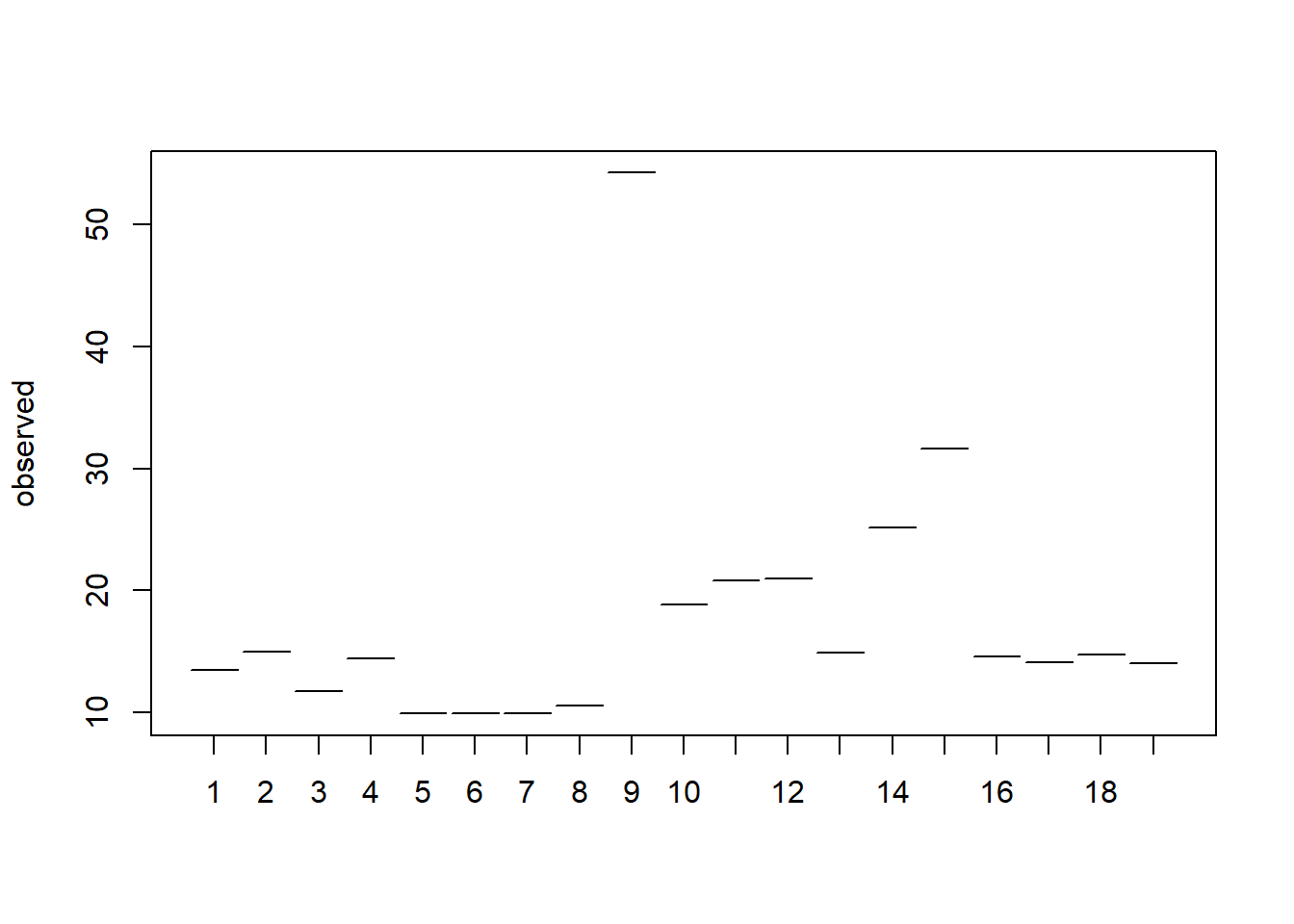
observed\_monthly <- decompose(observed)

plot(observed\_monthly)



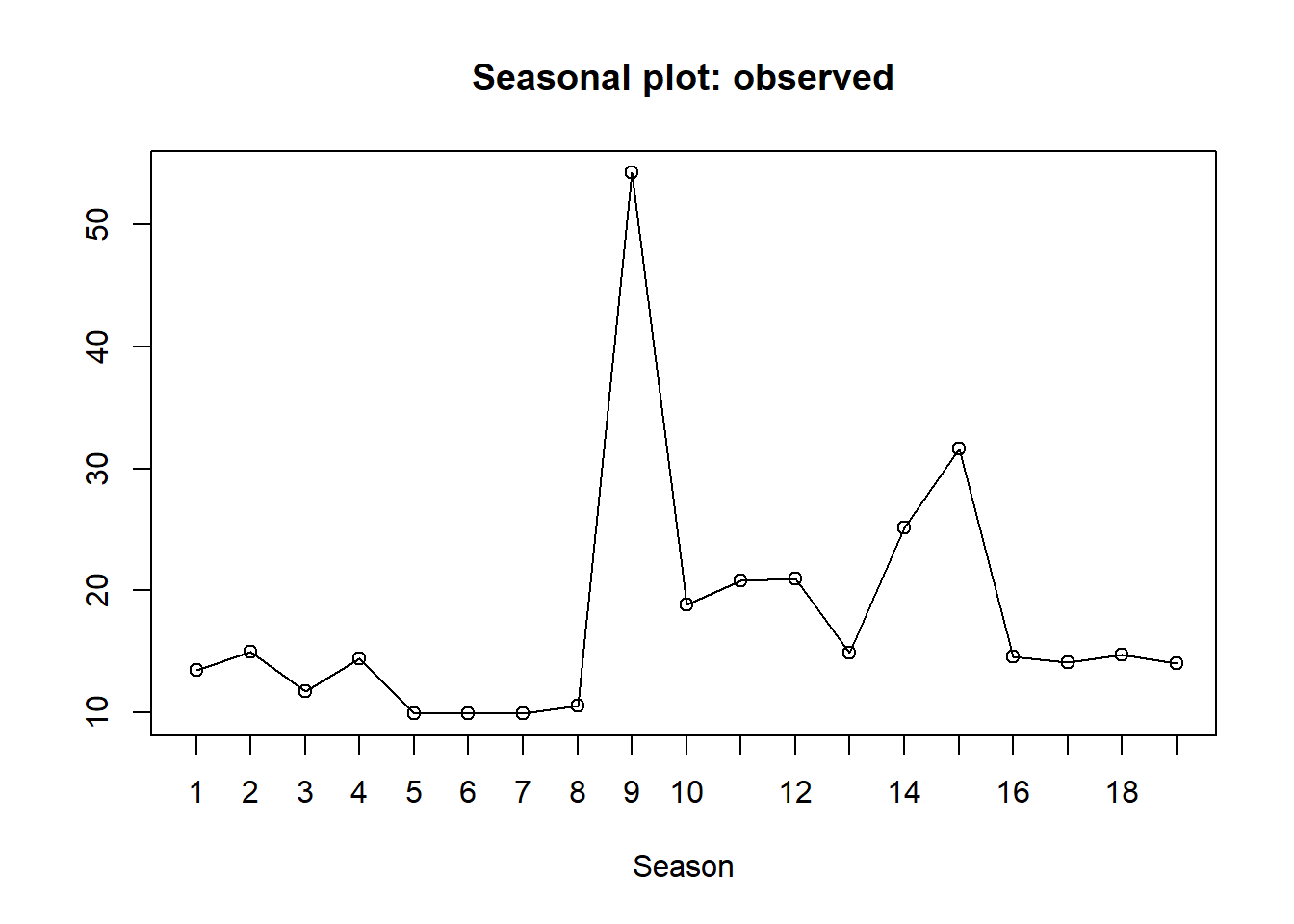
*#The seasonality is plotted*

monthplot(observed)



**library**(forecast)

seasonplot(observed)



*#Forecasting the duration of the bikes*

my\_timeseries <- data.frame(duration = observed, as.numeric(time(observed)))

names(my\_timeseries) <- c("duration", "time")

Mytimeseries <- tslm(duration~season+trend,my\_timeseries)

Forcast\_timeseries <- forecast(Mytimeseries,h=20)

autoplot(Forcast\_timeseries)

