Assignment2

R V Abhishek

2025-09-12

```r
# Loading necessary libraries
library(ggplot2)
library(zoo)
```

```
Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

```r
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
# Data initialization
values <- c(15,18,14,20,16,19,15,21,17,21,16,22,14,20,13,24,15,17,18,26)

# Mean
mean_val <- mean(values)
mean_val
```

```
[1] 18.05
```

```r
# Median
median_val <- median(values)
median_val
```

```
[1] 17.5
```

```r
# Mode function
get_mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
mode_val <- get_mode(values)
mode_val
```

```
[1] 15
```

Interpretation: The mean = r mean_val, median = r median_val, and the mode = r mode_val. These show the central tendency of quarterly sales.

```r
# Population and Sample calculations
pop_var  <- mean((values - mean_val)^2)
pop_sd   <- sqrt(pop_var)

samp_var <- var(values)
samp_sd  <- sd(values)

pop_var; pop_sd
```

```
[1] 11.8475
```

```
[1] 3.44202
```

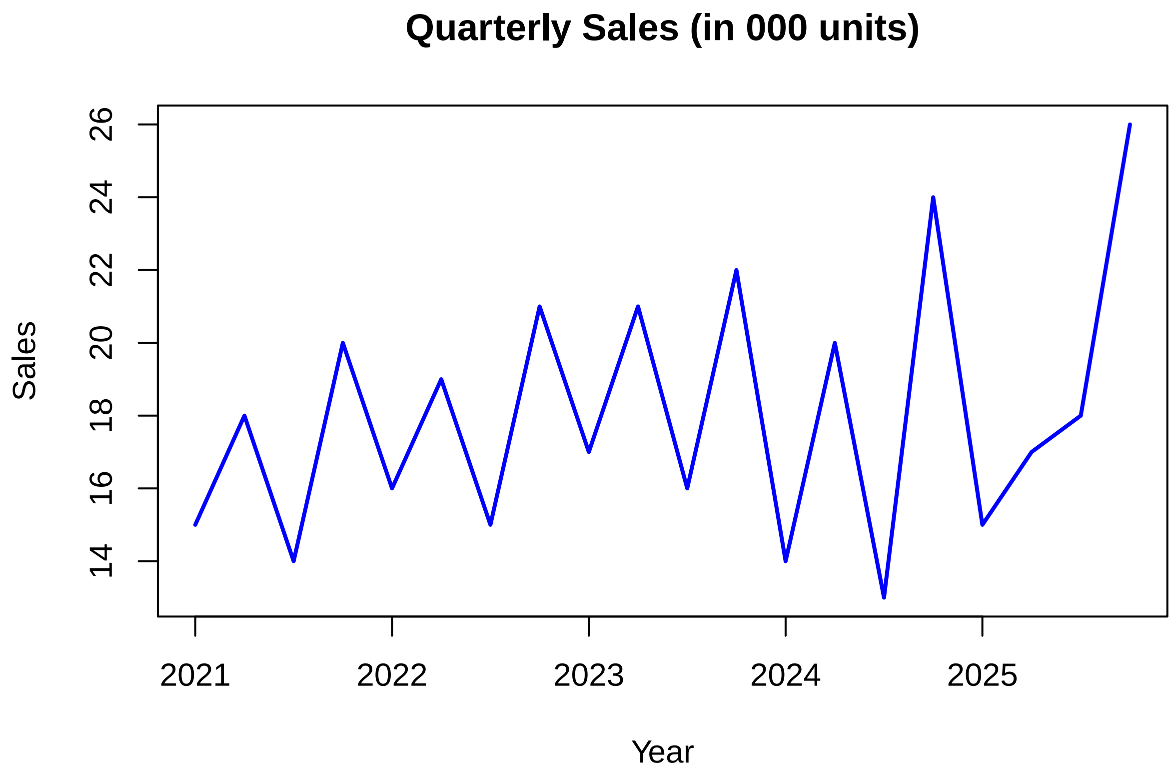
```r
samp_var; samp_sd
```

```
[1] 12.47105
```

```
[1] 3.531438
```

Interpretation: The population variance = r pop_var and standard deviation = r pop_sd. The sample variance = r samp_var and standard deviation = r samp_sd. This indicates the spread of quarterly sales data.

```r
# Time series Plot
tsq <- ts(values, start = c(2021,1), frequency = 4)
plot(tsq, col="blue", lwd=2, main="Quarterly Sales (in 000 units)",
     xlab="Year", ylab="Sales")
```



Comment: From the plot, sales are generally increasing in Q4 each year. Q1 tends to be the lowest. This indicates seasonality.

```
# Correlation value and optimal significance test
q1 <- c(15,16,17,14,15)
q2 <- c(18,19,21,20,17)

cor_val <- cor(q1, q2)
cor_val
```

```
[1] 0.4160251
```

```
cor.test(q1, q2)
```

```

	Pearson's product-moment correlation

data:  q1 and q2
t = 0.79241, df = 3, p-value = 0.486
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.7366105  0.9497065
sample estimates:
      cor 
0.4160251 
```

Hypotheses: H0: No correlation between Q1 and Q2 H1: There is correlation

Interpretation: The correlation = r round(cor_val,2). The p-value tells us if this correlation is significant.

```r
# Paired t-test
q4 <- c(20,21,22,24,26)
t.test(q1, q4, paired = TRUE)
```

```

    Paired t-test

data:  q1 and q4
t = -5.3079, df = 4, p-value = 0.006055
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -10.966153  -3.433847
sample estimates:
mean difference
           -7.2
```

Hypotheses: H0: Mean(Q1) = Mean(Q4) H1: Mean(Q1) Mean(Q4)

Conclusion: The p-value shows whether Q1 and Q4 sales are significantly different. Since Q4 values look much higher, we expect rejection of H0.

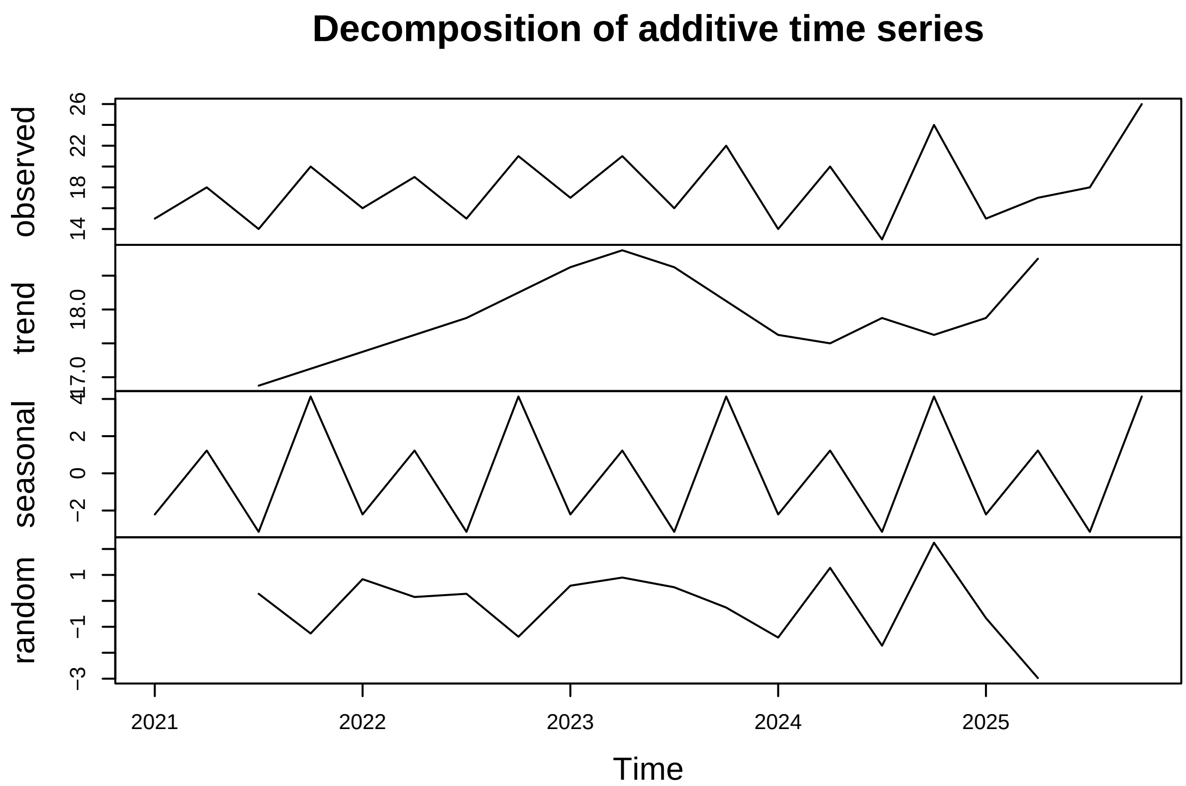
```r
# One sample t-test
t.test(values, mu = 18)
```

```

    One Sample t-test

data:  values
t = 0.063319, df = 19, p-value = 0.9502
alternative hypothesis: true mean is not equal to 18
95 percent confidence interval:
 16.39724 19.70276
sample estimates:
mean of x
    18.05
```

Hypotheses: H0: Mean quarterly sales = 18 H1: Mean quarterly sales 18

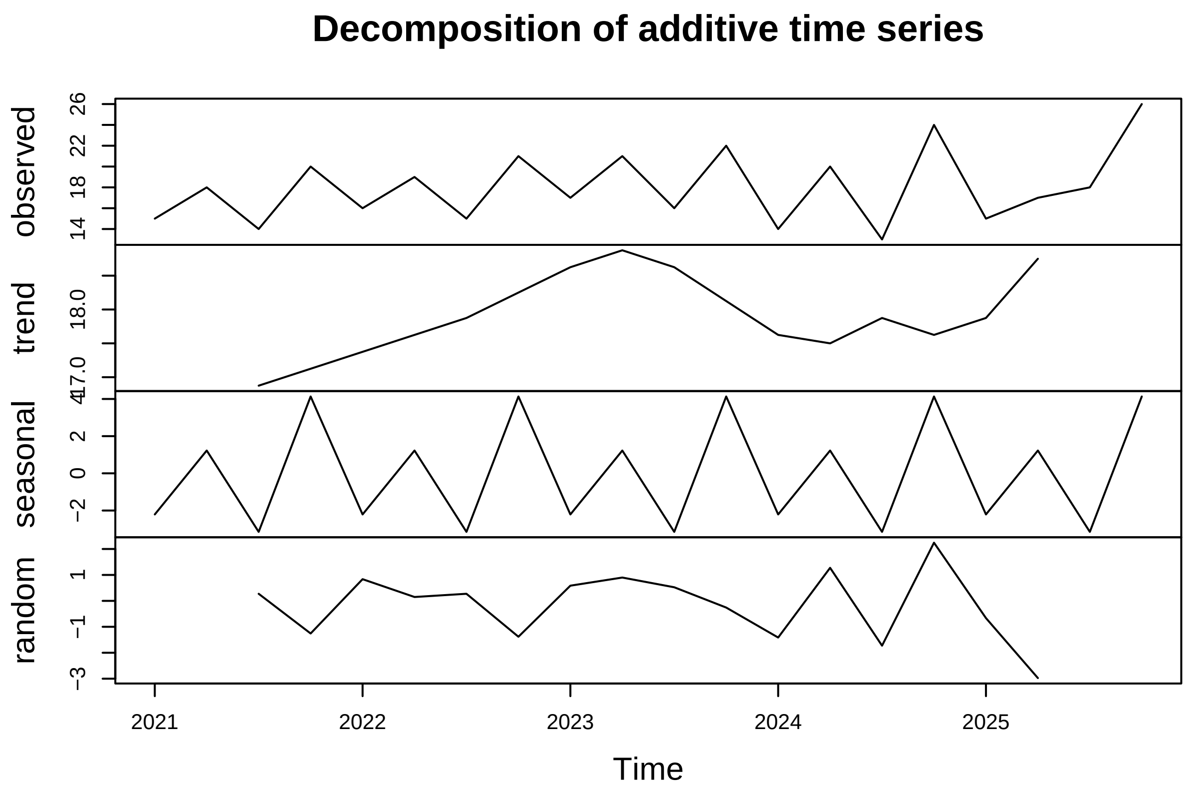
Conclusion: The test result shows if the average is statistically different from 18.

```r
# Additive Decomposition
dec_add <- decompose(tsq, type = "additive")
plot(dec_add)
```



Comment: The additive model splits sales into trend, seasonal and residual. We see upward spikes in Q4 consistently, so seasonality is strong.

```r
# Multiplicative Decomposition
dec_add <- decompose(tsq, type = "additive")
plot(dec_add)
```



Comment: The multiplicative decomposition also shows a seasonal effect. By comparing plots, multiplicative may fit better if the seasonal effect grows with trend, otherwise additive is fine.

```
# Preparation of data
df <- data.frame(
  year = rep(2021:2025, each = 4),
  quarter = rep(paste0("Q",1:4), times = 5),
  value = values
)

# Ratio to Moving Average Seasonal Indices
ma4 <- rollmean(values, k = 4, align = "center", na.pad = TRUE)
df_ma <- df %>% mutate(ma = ma4, ratio = value/ma)

seasonal_rtma <- df_ma %>%
  filter(!is.na(ratio)) %>%
  group_by(quarter) %>%
  summarise(avg_ratio = mean(ratio))

seasonal_rtma$norm <- seasonal_rtma$avg_ratio / mean(seasonal_rtma$avg_ratio)
seasonal_rtma
```

```
# A tibble: 4 x 3
  quarter avg_ratio  norm
  <chr>       <dbl> <dbl>
1 Q1          0.861 0.870
```

```
2 Q2          1.05  1.07
3 Q3          0.814 0.823
4 Q4          1.23  1.24
```

Comment: These are the seasonal indices using RTMA. They show how much each quarter deviates from the average trend.

```r
# Ratio to trend Seasonal Indices
time_index <- 1:length(values)   # create time variable

model <- lm(values ~ time_index) # fit trend line
trend <- predict(model)

df_tr <- tibble(
  value = values,
  trend = trend,
  quarter = cycle(tsq)
)

df_tr <- df_tr %>% mutate(ratio = value / trend)

season_rt <- df_tr %>%
  group_by(quarter) %>%
  summarise(avg_ratio = mean(ratio))

season_rt$norm <- season_rt$avg_ratio / mean(season_rt$avg_ratio)
season_rt
```

```
# A tibble: 4 x 3
  quarter avg_ratio  norm
    <dbl>     <dbl> <dbl>
1       1     0.869 0.869
2       2     1.06  1.06
3       3     0.838 0.838
4       4     1.23  1.23
```

Comment: These are the seasonal indices from the ratio-to-trend method. By comparing with Q9, the values are similar, confirming the seasonal pattern.