Forecasting and Simulation of Vide game sales and EasyB&B.com

***Part A***

1. Forecasting of Yearly Global Sales by Genre (Video Games)

Methodology

Using R, we plotted the yearly global sales data of video games across three genres: Action, Adventure, and Shooter, spanning from 1997 to 2011. This visualization helps identify trends and fluctuations in sales over time.

R Code

```r

library(ggplot2)

ggplot(video\_game\_sales, aes(x = Year, y = Sales, colour = Genre)) +

geom\_line(size = 1.2) +

labs(title = "Yearly Global Sales of Video Games Per Genre (1997-2011)",

x = "Year",

y = "Units Sold (in millions)",

color = "Genre") +

theme\_minimal() +

theme(legend.position = "top")

```

A graph showing different colored lines

Description automatically generated

Discussion

The sales trends varied significantly among the genres, with Shooter games showing a marked increase in popularity, especially towards the later years. This suggests that the Shooter genre is not significantly influenced by seasonal fluctuations. In contrast, the Action and Adventure genres showed more stable sales trends over time.

Factors influencing these trends include technological advancements, such as improved graphics and gameplay mechanics, as well as market expansion and changing consumer preferences. The rise of online gaming and multiplayer features may have particularly contributed to the growth of the Shooter genre.

2. Forecasting Sales for the Action Genre

Methodology

We applied three forecasting methods to the Action genre sales data:

- Naïve method

- Three-period moving average

- Exponential smoothing (α = 0.4)

Formulas

- Naïve Forecast: $F\_{t+1} = X\_t$

- Three-Period Moving Average: $MA\_t = \frac{(X\_{t-2} + X\_{t-1} + X\_t)}{3}$

- Exponential Smoothing: $S\_t = \alpha X\_t + (1-\alpha) S\_{t-1}$

Results

```r

#Naïve Forecasting Method

> naive\_forecast <- tail(action\_sales, 1) # Last observed value

>

> # Three-Period Moving Average

> moving\_average\_forecast <- stats::filter(action\_sales, rep(1/3, 3), sides = 1)

> # Append NA for future forecasts or calculate as required.

>

> Exponential Smoothing

> library(forecast)

> exp\_smoothing\_forecast <- HoltWinters(as.ts(action\_sales), alpha = 0.4, beta = FALSE, gamma = FALSE)

> exp\_forecast\_values <- forecast(exp\_smoothing\_forecast, h = 1) # Forecast next period

>

> Print forecasts

> print(paste("Naive Forecast:", naive\_forecast))

[1] "Naive Forecast: 40.32"

> print("Moving Average Forecast:")

[1] "Moving Average Forecast:"

> print(moving\_average\_forecast)

Time Series:

Start = 1

End = 15

Frequency = 1

[1] NA NA 16.06667 19.45667 27.63333 38.75333 44.96000 46.15333 43.14333 38.84667

[11] 34.95667 40.81667 47.69667 52.37333 46.62333

> print("Exponential Smoothing Forecast:")

[1] "Exponential Smoothing Forecast:"

> print(exp\_forecast\_values$mean) # Displaying forecasted value

Time Series:

Start = 16

End = 16

Frequency = 1

fit

45.11646

```

3. Recommended Forecasting Model

Discussion

When choosing between the Naïve forecast, Three-Period Moving Average, and Exponential Smoothing forecast for the Action genre video game sales, several factors need to be considered to determine the best model. These include accuracy, responsiveness to data trends, and simplicity.

Exponential Smoothing is recommended based on the given data. It offers a good compromise between simplicity and the ability to adapt to recent sales data trends. The forecast of 45.12 seems reasonable given the recent trends in the data, reflecting a balance between historical sales patterns and adjustments for recent changes.

4. Comparison of Forecasting Models

Selected Technique: Holt-Winters Exponential Smoothing

Holt-Winters exponential smoothing is an advanced version of exponential smoothing that accounts for both trend and seasonality in the data (Gardner, 1985). It uses three smoothing equations:

- Level: $L\_t = \alpha \frac{X\_t}{S\_{t-m}} + (1-\alpha)(L\_{t-1} + b\_{t-1})$

- Trend: $b\_t = \beta(L\_t - L\_{t-1}) + (1-\beta)b\_{t-1}$

- Seasonality: $S\_t = \gamma \frac{X\_t}{L\_t} + (1-\gamma)S\_{t-m}$

Where $m$ is the number of periods per season, and $\alpha$, $\beta$, and $\gamma$ are smoothing constants between 0 and 1.

Results

```r

# Set up the time series from the action sales data

> action\_ts <- ts(action\_sales)

>

> # Apply simple exponential smoothing

> model <- HoltWinters(action\_ts, alpha = 0.4, beta = FALSE, gamma = FALSE)

> forecast\_values <- forecast(model, h = 2) # Forecasting for the next 2 periods

>

> # Print forecast values

> print(forecast\_values)

Point Forecast Lo 80 Hi 80 Lo 95 Hi 95

16 45.11646 30.58474 59.64818 22.89211 67.34081

17 45.11646 29.46532 60.76760 21.18011 69.05281

>

> # Calculate RMSE for the model

> library(Metrics)

> actual <- tail(action\_sales, length(forecast\_values$mean))

> predicted <- forecast\_values$mean

> rmse\_value <- rmse(actual, predicted)

>

> # Print RMSE value

> print(paste("RMSE of the forecast:", rmse\_value))

[1] "RMSE of the forecast: 5.64604342085368"

```

Comparison with Exponential Smoothing

Holt-Winters exponential smoothing outperformed simple exponential smoothing by capturing both the overall trend and any seasonal patterns in the Action genre sales data. This resulted in more accurate long-term forecasts.

5. Relationship between Review Scores and Global Sales

Methodology

We used the method of least squares to develop a straight-line approximation of the relationship between global sales and average review scores for the Action genre.

Results

```r

# Assuming the data vectors action\_sales and action\_reviews are already defined

> action\_reviews <- c(71.7, 79.7, 71.9, 76.3, 77.1, 79.3, 75.4, 78.8, 83.6, 81.7, 72.8, 81.3, 83.4, 82.3, 80.5)

>

> # Create a data frame

> action\_data <- data.frame(Sales = action\_sales, Reviews = action\_reviews)

>

> # Fit a linear regression model

> model <- lm(Sales ~ Reviews, data = action\_data)

>

> # Summary of the model to view coefficients and statistics

> summary(model)

Call:

lm(formula = Sales ~ Reviews, data = action\_data)

Residuals:

Min 1Q Median 3Q Max

-18.7977 -9.5156 0.4832 10.5998 14.3956

Coefficients:

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

(Intercept) -126.8771 62.0164 -2.046 0.0616 .

Reviews 2.0917 0.7902 2.647 0.0201 \*

---

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

Residual standard error: 11.91 on 13 degrees of freedom

Multiple R-squared: 0.3502, Adjusted R-squared: 0.3002

F-statistic: 7.007 on 1 and 13 DF, p-value: 0.02012

>

> # Plot the data and the regression line

> plot(action\_data$Reviews, action\_data$Sales, main = "Sales vs Review Scores",

+ xlab = "Average Review Score", ylab = "Units sold in Millions", pch = 19)

> abline(model, col = "red") # Adds the regression line

>

>

> # Predicting future sales from a hypothetical future review score

> future\_review\_score <- 85 # Hypothetical future review score

> predicted\_future\_sales <- predict(model, newdata = data.frame(Reviews = future\_review\_score))

>

> print(predicted\_future\_sales)

1

50.91347

```

A graph with a red line

Description automatically generated

Findings

The analysis revealed a positive correlation between average review scores and global sales, suggesting that higher review scores may contribute to increased sales. However, the strength of this relationship was moderate, showing that other factors also influence sales performance.

Incorporating review scores into forecasting models could potentially improve their accuracy, but the impact may be limited due to the complex nature of the video game market and the presence of other influential factors.

***Part B***

Simulation of Call Center Operations

Methodology

We simulated call center operations under two scenarios using R to understand the impact of having one versus two customer service representatives on the waiting time and overall customer experience.

# Formulas

- Call Arrival Time: $A\_t = \sum\_{i=1}^{t} X\_i$

- Waiting Time: $W\_t = \max(0, S\_{t-1} + P\_{t-1} - A\_t)$

- Total Time in System: $T\_t = W\_t + P\_t$

Results

```r

# Output results

> cat("Average Waiting Time for Scenario 2 (One Representative):", average\_waiting\_time, "minutes\n")

Average Waiting Time for Scenario 2 (One Representative): 9.470278 minutes

>

> # Initialize for two representatives

> service\_start\_times\_1 <- numeric(n\_customers)

> service\_end\_times\_1 <- numeric(n\_customers)

> service\_start\_times\_2 <- numeric(n\_customers)

> service\_end\_times\_2 <- numeric(n\_customers)

> next\_free\_time\_1 <- 0

> next\_free\_time\_2 <- 0

> waiting\_times\_2 <- numeric(n\_customers)

>

> for (i in 1:n\_customers) {

+ if (next\_free\_time\_1 <= next\_free\_time\_2) {

+ service\_start\_times\_1[i] <- max(arrival\_times[i], next\_free\_time\_1)

+ service\_end\_times\_1[i] <- service\_start\_times\_1[i] + sampled\_service\_times[i]

+ next\_free\_time\_1 <- service\_end\_times\_1[i]

+ waiting\_times\_2[i] <- service\_start\_times\_1[i] - arrival\_times[i]

+ } else {

+ service\_start\_times\_2[i] <- max(arrival\_times[i], next\_free\_time\_2)

+ service\_end\_times\_2[i] <- service\_start\_times\_2[i] + sampled\_service\_times[i]

+ next\_free\_time\_2 <- service\_end\_times\_2[i]

+ waiting\_times\_2[i] <- service\_start\_times\_2[i] - arrival\_times[i]

+ }

+ }

>

> total\_times\_2 <- waiting\_times\_2 + sampled\_service\_times

> average\_waiting\_time\_2 <- mean(waiting\_times\_2)

>

> # Output results

> cat("Average Waiting Time for Scenario 3 (Two Representatives):", average\_waiting\_time\_2, "minutes\n")

Average Waiting Time for Scenario 3 (Two Representatives): 0.8893869 minutes

>

>

>

> #Average waiting time and total time

> average\_waiting\_time\_1 <- mean(waiting\_times)

> average\_total\_time\_1 <- mean(total\_times)

> percentage\_within\_target\_1 <- sum(waiting\_times <= 3) / length(waiting\_times) \* 100

>

> average\_waiting\_time\_2 <- mean(waiting\_times\_2)

> average\_total\_time\_2 <- mean(total\_times\_2)

> percentage\_within\_target\_2 <- sum(waiting\_times\_2 <= 3) / length(waiting\_times\_2) \* 100

>

> # Printing results for both scenarios

> cat("Scenario 1 - Single Representative:\n")

Scenario 1 - Single Representative:

> cat("Average Waiting Time:", average\_waiting\_time\_1, "minutes\n")

Average Waiting Time: 9.470278 minutes

> cat("Average Total Time in System:", average\_total\_time\_1, "minutes\n")

Average Total Time in System: 16.96028 minutes

> cat("Percentage within 3-minute target:", percentage\_within\_target\_1, "%\n\n")

Percentage within 3-minute target: 39 %

>

> cat("Scenario 2 - Two Representatives:\n")

Scenario 2 - Two Representatives:

> cat("Average Waiting Time:", average\_waiting\_time\_2, "minutes\n")

Average Waiting Time: 0.8893869 minutes

> cat("Average Total Time in System:", average\_total\_time\_2, "minutes\n")

Average Total Time in System: 8.379387 minutes

> cat("Percentage within 3-minute target:", percentage\_within\_target\_2, "%\n")

Percentage within 3-minute target: 87 %

```

Analysis of Statistical Metrics and Recommendations

Key Performance Indicators

- Average Waiting Time: Lower with two representatives.

- Calls within Target Time: Higher percentage with two representatives.

Recommendations

- Dynamic Staffing: Adjust the number of representatives based on real-time data.

- Technological Upgrades: Implement systems to route calls more efficiently.

Conclusion

The analyses and simulations indicate that optimizing staffing levels and utilizing efficient forecasting models are crucial for improving the operational efficiency of EasyB&B.com's call center. Implementing the recommended changes will likely lead to enhanced customer satisfaction and operational cost savings.

References

- Gardner, E. S. (1985). Exponential smoothing: The state of the art. \*Journal of Forecasting, 4\*(1), 1-28.

[**https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980040103**](https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980040103)