



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

**COST ESTIMATION AND ANALYSIS
CAPSTONE PROJECT**

**Optimal Portfolio Decision Making
Under Cost Uncertainty & Learning Curve Application**

by

PAULA BRUM
SYLVIA MEDELLIN
JARED PETCHEL

Advisor:

CAPT Douglas E. Otte USN(Ret.)

THIS PAGE INTENTIONALLY LEFT BLANK

**Optimal Portfolio Decision Making
Under Cost Uncertainty & Learning Curve Application**

Team Members

Paula Brum
Bachelor's – Industrial Engineering
Universidade Federal Fluminense, Niterói - RJ

Sylvia Medellin
Bachelor's – Accounting
Our Lady of the Lake University, San Antonio, TX

Jared Petchel
Bachelor's – Secondary Mathematics Education
Master's - Mathematics
University of Pittsburgh, Pittsburgh, PA

Advisor:

CAPT Douglas E. Otte USN(Ret.)
Naval Post-Graduate School

Submitted in partial fulfillment of the
requirements for the degree of

MASTER'S OF COST ESTIMATION AND ANALYSIS

from the

**NAVAL POSTGRADUATE SCHOOL
25 March 2022**

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

| | |
|--|-----|
| LIST OF TABLES..... | iv |
| LIST OF FIGURES..... | iv |
| LIST OF ACRONYMS AND ABBREVIATIONS | vi |
| I – EXECUTIVE SUMMARY | vii |
| PROBLEM STATEMENT..... | vii |
| PROPOSED SOLUTION..... | vii |
| FINDINGS & CONCLUSIONS..... | vii |
| RECOMMENDATION | vii |
| II – STUDY INTRODUCTION | 1 |
| A – INTRODUCTION..... | 1 |
| B – BACKGROUND..... | 2 |
| 1 – PURPOSE | 2 |
| 2 – SCOPE | 2 |
| C – ASSUMPTIONS AND CONSTRAINTS | 2 |
| 1 – ASSUMPTIONS..... | 2 |
| 2 – CONSTRAINTS..... | 3 |
| III – ANALYTICAL APPROACH..... | 4 |
| A – METHODOLOGY | 4 |
| B – MEASURES OF EFFECTIVENESS | 5 |
| IV – ANALYSIS..... | 7 |
| A – THEORETICAL NON-LINEAR PROGRAMMING MODEL DESCRIPTION | 7 |
| B – ANALYTICAL TOOL..... | 7 |
| C – MODULE ONE: REGRESSIONS | 8 |
| 1 – FRIGATES REGRESSIONS AND COST EQUATIONS..... | 8 |
| 2 – PATROLS REGRESSIONS AND COST EQUATIONS..... | 13 |
| 3 – SUBMARINES REGRESSIONS AND COST EQUATIONS | 17 |
| D – MODULE TWO: MONTE CARLO SIMULATION | 20 |
| 1 – FRIGATES COST DISTRIBUTION..... | 22 |
| 2 – PATROLS COST DISTRIBUTION | 23 |
| 3 – SUBMARINES COST DISTRIBUTION | 24 |
| E – MODULE THREE: NON-LINEAR PROGRAMMING MODEL | 25 |
| 1 - THEORETICAL NON-LINEAR PROGRAMMING MODEL..... | 25 |

| | |
|---|----|
| 2 – ACCOUNTING FOR LEARNING CURVES WITH A MULTI-VARIABLE MODEL..... | 26 |
| 3 – TRANSLATING THE THEORETICAL NLP MODEL INTO R© | 27 |
| 4 – SPONSOR-SUGGESTED ALTERATION..... | 28 |
| V – CONCLUSION..... | 31 |
| Appendix A: Regressions Programming Script..... | 32 |
| 1 Frigates Regression | 32 |
| 2 Patrols Regression..... | 34 |
| 3 Submarines Regression | 36 |
| Appendix B: Monte Carlo Simulation Programming Script..... | 39 |
| 1 Frigates Monte Carlo Simulation | 39 |
| 2 Patrols Monte Carlo Simulation..... | 40 |
| 3 Submarines Monte Carlo Simulation | 42 |
| Appendix C: Non-Linear Programming Script | 44 |
| 1 Minimum Budget | 44 |
| 2 Fixed Budget..... | 46 |
| Appendix D: Reference Tables for Data Array | 52 |
| 1 – Frigates Data | 52 |
| 2 – Patrols Data..... | 53 |
| 3 – Submarines Data | 54 |
| References | 55 |

LIST OF TABLES

| | |
|---|----|
| Table 1: Frigates Data | 9 |
| Table 2: Frigates Regressions Comparison | 11 |
| Table 3: Patrols Data | 13 |
| Table 4: Patrol Regressions Comparison..... | 15 |
| Table 5: Submarines Data | 17 |
| Table 6: Submarines Regression Comparison..... | 18 |
| Table 7: Values and learning Curves used in NLP Model | 27 |

LIST OF FIGURES

| | |
|--|---|
| Figure 1: Modules and Integration | 5 |
| Figure 2: Frigates Correlation Matrix..... | 9 |

| | |
|---|----|
| Figure 3: Frigates Regression Results..... | 10 |
| Figure 4: Frigates Model 7 Regression – Residual Normality | 11 |
| Figure 5 Frigates Model 7 - Heteroskedasticity | 11 |
| Figure 6: Frigates Final Regression..... | 12 |
| Figure 7: Frigates Regression Chart | 12 |
| Figure 8: Patrols Correlation Matrix | 13 |
| Figure 9: Patrols Regressions Results..... | 14 |
| Figure 10: Patrols Model 3 Regression – Residual Normality..... | 15 |
| Figure 11: Patrols Model 3 Regression - Heteroskedasticity | 15 |
| Figure 12: Patrols Final Regression | 16 |
| Figure 13: Patrols VIF Test Results | 16 |
| Figure 14: Submarines Correlation Matrix..... | 17 |
| Figure 15: Submarines Regressions Results..... | 18 |
| Figure 16: Model 7 Regression – Residual Normality | 19 |
| Figure 17: Submarines Model 7 - Heteroskedasticity..... | 19 |
| Figure 18: Submarines Regression Chart | 20 |
| Figure 19: SME Assessment for Frigates Displacement..... | 22 |
| Figure 20: Frigates Cost Cumulative Distribution Function | 22 |
| Figure 21: Frigates Cost Distribution Function – Main Parameters and Comparison | 22 |
| Figure 22: SME Assessment for Patrols Displacement and Length | 23 |
| Figure 23: Patrols Cost Cumulative Distribution Function..... | 23 |
| Figure 24: Patrols Cost Distribution Function – Main Parameters and Comparison | 23 |
| Figure 25: SME Assessment for Submarine Displacement | 24 |
| Figure 26: Submarines Cost Cumulative Distribution Function | 24 |
| Figure 27: Submarines Cost Distribution Function – Main Parameters and Comparison | 24 |
| Figure 28: Total Cost of Each Ship and Their Respective Shadow Price | 29 |
| Figure 29: Shadow Prices Relative to the Remaining Budget..... | 30 |
| Figure 30: Percentage of Each Ship Type Purchasable with Remaining Budget..... | 30 |

LIST OF ACRONYMS AND ABBREVIATIONS

[CER](#): Cost Estimating Relationship

[CST](#): Cost

[CV](#): Coefficient of Variation

[DIS](#): Displacement

[DoD](#): Department of Defense

[GRET](#): Software (Gnu Regression, Econometrics and Time-series Library)

[JMP](#): Software (John's Macintosh Project)

[LEN](#): Length

[MOE](#): Measures of Effectiveness

[NLP](#): Non-Linear Programming Model

[POC](#): Point of Contact

[RNG](#): Range

[SER](#): Schedule Estimating Relationship

[SME](#): Subject Matter Expert

[U.S.](#): United States of America

[VIF](#): Variance Inflation Factor

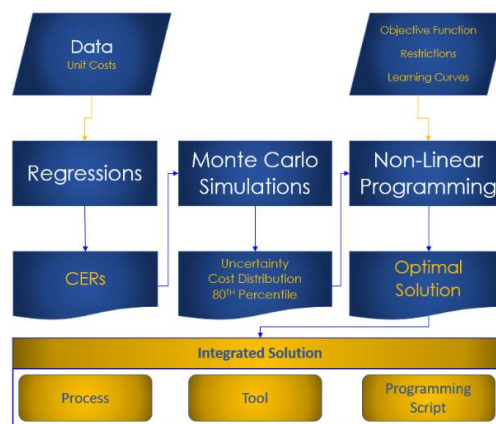
I – EXECUTIVE SUMMARY

PROBLEM STATEMENT

Determine optimal portfolio output using the costs of different systems. Given those systems' costs are not static, due to uncertainty, learning curve effects, etc.; the optimization “answers” should reflect the dynamics of the cost calculations.

PROPOSED SOLUTION

Cost estimators are involved in a significant amount of portfolio analysis and funding constraints. This team combines various cost estimating tools that provide portfolio optimization while accounting for uncertainty and learning curves. The new process, depicted in the process flow below, provides a way to maximize portfolio execution. It integrates Linear Regressions, Monte Carlo Simulations, and Non-Linear Programming through code in R© software language to produce an optimized solution. Cost estimators can now perform more in-depth analysis of their portfolio and provide recommendations to decision makers on how to manage scarce resources. This new approach helps determine the best combination of Systems Acquisition elements.



FINDINGS & CONCLUSIONS

The team determined it is possible to combine many aspects of cost estimating to achieve an optimal answer for multiple systems' acquisition. The process combines various cost estimating tools which results in an integrated solution with in-depth analysis. R© is an open-source software that performs concurrent statistical analysis of large amounts of data. Cost estimators can use their own data and apply it to the coding package created for this process. Thus, the process is flexible enough to add more complexity like accounting for Schedule Estimating Relationships, Manpower costs, or other cost elements within the project WBS. The building blocks are there for other cost estimators to make this process fit their needs.

RECOMMENDATION

The team recommends Acquisition leadership and cost estimators begin using this tool. Doing so will provide more in-depth and variable analysis within the decision-making process. Using this process results in the best combination of systems' acquisition with in-depth statistical analysis and gives leadership the optimal solution to allocate scarce resources.

II – STUDY INTRODUCTION

A – INTRODUCTION

Cost estimators working for the United States Federal Government are required to complete various courses in pursuit of a certificate to ensure they learn the regulatory requirements of cost estimating and federal acquisition. Because funding is a major constraint on government budgets, cost estimators are involved in a significant amount of portfolio allocation and what-if analyses to optimize acquisitions.

Despite all the required courses, cost estimators are still confined to a limited number of stove-piped commercial software to do statistical analysis. In most instances, cost estimators confine themselves to Microsoft Excel, ACEIT, or wInsight when performing data analysis. However, because of their commercial nature it is difficult to manipulate the pre-written code. These software programs also have a limited capacity for data analysis.

There are more robust software programs that can handle data analysis better than these systems. Some must be purchased, while others are free for public use. These software systems are available to use today but are not widely used due to several factors. A couple roadblocks are lack of programming experience and cost of software licenses. The DoD would have to perform a cost benefit analysis to determine if the cost of teaching programming to cost estimators is beneficial to the government. Another benefit analysis is needed to determine if purchasing new software packages would improve cost estimates.

This team's assignment is to create a solution that determines the optimal portfolio output using the unit costs of different systems (made up of their constituent parts) in a situation where those systems' costs are not static, due to uncertainty and learning curve effects. The optimal solution should reflect the dynamics of the cost calculations. Said another way, how can cost estimators determine the best combination of commodities that will use resources in the most efficient way possible?

The team created a solution to provide a process associated with tools (software) and models capable of meeting the requirement, i.e., the optimized answers reflect the dynamics of the cost calculations. Cost estimators can be limited in options when providing alternatives to acquisition leadership. The team wants to give estimators a way to find alternatives that provide more insight into data analysis and produce an optimized answer. This paper details the challenge of producing a cost model that accounts for uncertainty, applies learning curves, and explores Cost Estimating Relationships (CERs). The team used publicly available data to produce cost equations; accounted for uncertainty by running the data through Monte Carlo Simulations; and developed a non-linear programming (NLP) model to produce an interim response to the problem. During the process, new tools were explored and used.

B – BACKGROUND

1 – PURPOSE

The purpose of this project is to give cost estimators a process road map that could be used in any portfolio analysis accounting for uncertainty and learning curves. The road map provides additional analysis options when leadership seeks recommendations on how best to use the limited funding resources allocated to a military agency like the U.S. Navy. Cost estimators are limited by software that cannot compute large amounts of data. Often, resource drills come down suddenly with short turnaround times. Currently, estimators run analyses on software like Excel, which has limited data statistical analysis capability and capacity. In Excel, statistical analysis is limited to the ‘Add-ins’ that come with the Microsoft Office software package. Additional statistical applications can be added like JMP or @Risk. However, these types of software licenses usually come at a significant cost. If an estimator could take several variables and run them through a defined process within a coding package, they would then be able to provide a viable answer with thorough statistical analysis behind it. Leadership then can have confidence they are using resources in the most cost-efficient way possible.

2 – SCOPE

This project is limited to using publicly available data, which has advantages and disadvantages. One advantage is the team can be creative in coming up with its own scope and solution. On the other hand, approaching such an open-ended problem seems like a disadvantage because the possibilities are endless. It becomes overwhelming to consider that any and every data set could be used. The sponsor advised to keep the initial problem small enough to keep data analysis manageable and then increase complexity as the model proved fruitful. The NLP is created with a simplified objective function, notional restrictions, and with minimal variables to test the process. Once that equation proves to produce acceptable results, then the NLP can be modified and made more complex to produce a realistic model.

The team sponsor suggested exploring Schedule Estimating Relationships (SERs) as well. However, it was agreed that the problem presented was complex enough with three factors. Thus, SERs are not included in this model.

A non-linear programming model was created using a hypothetical situation involving naval defense systems like frigates, patrol vessels, and submarines. Cost and technical data about the systems were researched through several websites containing publicly available data. Some data came directly from various countries’ naval defense websites. Most data were obtained through sites that keep historical naval information for countries around the world.

C – ASSUMPTIONS AND CONSTRAINTS

1 – ASSUMPTIONS

As mentioned previously in the scope section, the model will only account for CERs and not SERs. This will reduce complicating the cost estimate further and the need for specialized and expensive software. To further simplify the data analytics, the model follows naval industry

standards for learning curves. NLP constraints were created by the team to limit the range of answers produced by the model. Lastly, the constraints were developed for demonstration purposes only and actual constraints would be influenced by overall operational portfolio demands.

2 – CONSTRAINTS

The sponsor requests the use of Linear Programming or Goal Programming to achieve an optimized answer. Learning curves are applied to the unit price to account for variances in unit price for systems purchased (Balakrishnan, et al. 2017). Given that applying learning curves to equations converts them to non-linear functions, the cost model created fits within NLP.

Additionally, this project did not come with an accompanying set of data. The team had to, in essence, create its own data array. The only information available for this project is publicly available data. Obtaining data points required extensive scouring of the internet for public information available for naval vessels. Since no data was provided, then no data constraints existed either. The problem provided by the U.S. Navy sponsor seemed open to a wide interpretation. The only way to approach this challenge is to limit the scope of the data and create a hypothetical situation to provide an objective function for the NLP model.

III – ANALYTICAL APPROACH

A – METHODOLOGY

The overall project was divided into phases. The first phase is Solution Conception, and it entailed group discussions and Subject Matter Experts (SME) involvement to help design the solution scope definition. Professors Gregory Mislick and Kevin Maher of the Naval Postgraduate School were consulted and assisted with regressions and linear programming insights. Justin Moul, U.S. Navy team sponsor, also assisted by clarifying the issue and project scope.

Phase two was Modules Development. This phase was inspired by how the brain masters a complex movement most efficiently, like dancing, for example. The brain divides the whole dance into smaller parts. It starts with a few moves and when they are under control, it combines them to a part of the dance. It keeps adding extra moves until the person can perform the full dance.

Similarly, the problem is divided up into modules. These modules were developed individually before connecting them into one overarching model. Three separate modules were created. The results of each module feeds into the next one. The first module objective is to create cost regressions and produce CERs for each system. The second module objective is to use Monte Carlo Simulation to account for uncertainty and produce cost distributions so the point estimate at the eightieth percentile is considered as an input for the NLP objective function. The third module objective is to develop an objective function and restrictions for an NLP model to achieve the optimal portfolio output. In this last module, the learning curves' effect were considered within the objective function of the NLP. Figure 1 shows how the process flows and how the modules are connected. At the end of the three modules' execution is an integrated solution that accounts for all the statistical elements produced within the modules.

The objective of this phase was to develop the solution for each module, including gathering data, choosing the appropriate tools, and producing a solution based on the data. The team created a data set based on various vessels from various countries. The data set is used to create a cost equation that is then used to run regressions, Monte Carlo Simulations, and ultimately the NLP model.

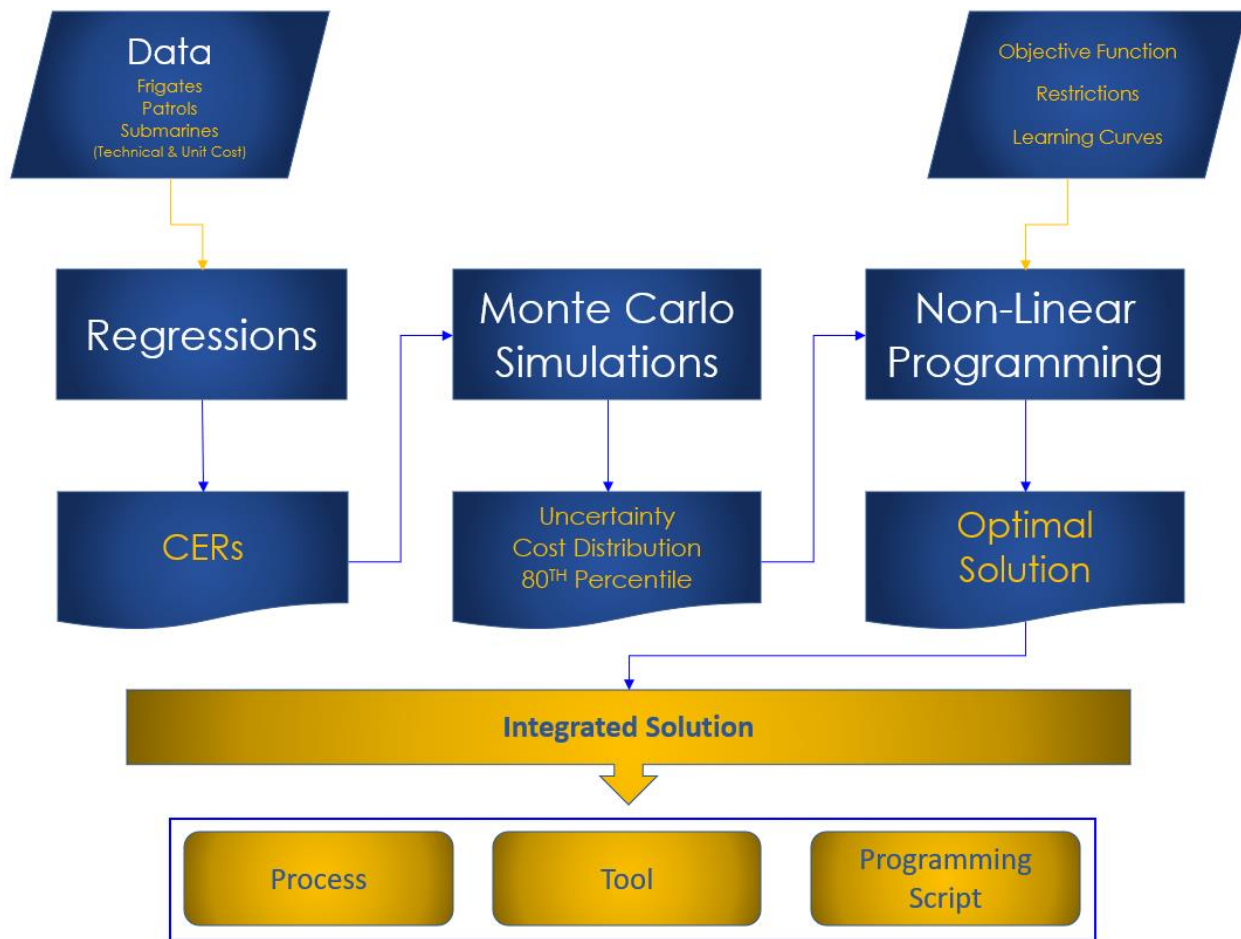


Figure 1: Modules and Integration

Phase three is Modules Integration. The phase involves adjusting the data and integrating the modules into one unique solution. The results from modules one and two were integrated into the third module, the NLP model.

Phase four is Process, Tools and Model Description. This report documents the entire process along with tools used in each module, the non-linear programming model, and the integrated solution. The appendices include the programming scripts. This report is the result of phase four and should be used as the guide for cost estimators to use with their own portfolio data.

This project goes beyond creating a credible and reliable cost estimate. The solution is designed to be used after a cost estimate has been created and after a unit price is determined.

B – MEASURES OF EFFECTIVENESS

The measures of effectiveness (MOEs) are metrics that will prove the attainment of the study objective. Module one involves regressions and CERs. The selection criteria used for candidate CERs is the Regression Hierarchy presented in Professor Mislick’s and Nussbaum’s

book, *Cost Estimation: Methods and Tools*. The hierarchy consists of two phases. The first part of phase one is to apply the commonsense test, which takes into consideration whether the slope of the regression line is realistic and does not refer to individual data points. If the regression passes this initial test, then the data must pass two additional criteria. The regression must produce an F-statistic and t-statistic significance (p-value) below 20%. A regression that does not meet all three criteria fails phase one and the regression must be rejected. If the regression passes through phase one, then it can be considered for phase two. The second phase compares those valid regressions and selects the best option based on the higher R-Square value, lower Standard error, and lower coefficient of variations (Mislick and Nussbaum 2015).

Other tests were applied to module one that check the regressions' required conditions and presence of multicollinearity in cases of multiple regressions. Those tests are called Regression Diagnostics. The tests to check the regressions' required conditions are checking if the error variable is normal and if the error variance is constant (i.e., homoskedasticity test). Finally, a multicollinearity test is applied to check if the independent variables are highly correlated. The cost equations that result from regressions and pass through the rigor of all these tests are used in module two, Monte Carlo Simulations.

IV – ANALYSIS

A – THEORETICAL NON-LINEAR PROGRAMMING MODEL DESCRIPTION

The sponsor wants the cost estimating team to determine the best way to allocate a certain budget between naval systems (frigates, patrol vessels, and submarines). The systems compete against each other for a portion of that budget. The cost model should help determine which combination of systems will maximize the use of that budget amount. The solution must also address the dynamics of the cost calculations such as, uncertainty and learning curves. The team developed constraints related to operational restrictions as listed below:

- The sum of submarines and frigates must be at least 16 units
- There must be at least one unit of each system
- There must be at least four patrols for each submarine
- There must be at least two frigates for each submarine
- There must be at least two patrols for every frigate

B – ANALYTICAL TOOL

The sponsor suggested the group use R© to steer away from using Excel, which has limitations when performing statistical analysis on large data sets. R©, developed at Bell Laboratories, is an integrated suite of software facilities for data manipulation, calculation, and graphical display (W.N. Venables 2021). His suggestion also comes from the fact that the software is free and has existing coding libraries developed that could be used for this project.

R© requires programming knowledge, and, in the beginning, it seemed the learning curve was too steep given the six months available to complete the project. However, through research and internet tutorials it becomes easier to use and understand the sponsor's recommendation for this tool. Among other things, R© has:

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either directly at the computer or on hardcopy, and
- a well-developed, simple, and effective programming language (called 'S') which includes conditionals, loops, user-defined recursive functions, and input and output facilities. (W.N. Venables 2021)

Additionally, R© is robust enough to handle the three modules. It became possible, by learning how to program in R©, to develop all modules integrated using a single tool. A major benefit of R© is the ability to store large amounts of data. This is crucial to the success of this project. A similar model calculated in Excel will only be able to store roughly 15,000 different portfolio configurations due to its limitations. While this may seem like a large amount, it limits

the number of ships that could be purchased within each different portfolio calculated. This creates the issue that the model may not accurately output the most optimal solution. Using R©, the team can bypass this limitation and consider 1,000,000 different portfolios. Furthermore, once the constraints are applied, the NLP model output is 17,892 possible solutions. This amount surpasses Excel's limitations. While the use of R© requires the team to overcome the learning curve to learn a new programming language; it also gives the team the ability to create a much more robust model.

C – MODULE ONE: REGRESSIONS

Module one's objective is to create cost regressions and produce cost equations for each system type. In this module, DoD prescribed cost estimating methods were used to achieve the results needed to advance to module two. The raw data was run through several regressions to determine which combination of independent variables produces the best cost equation, statistically.

The programming script is available in Appendix A: Regressions Programming Script. Cost estimators can use the R© script in case of facing the same challenge and only requires minor adaptations to reflect the specifics of the problem statement. Also, cost estimators can use the script when facing an inordinate amount of data, since Excel has limitations and R© is a more robust software.

1 – FRIGATES REGRESSIONS AND COST EQUATIONS

The data for the frigates' regression, consists of cost and technical information like displacement, range, and length. Table 1 shows the data array. The complete list with sources is listed in Appendix D, '1 – Frigates Data'.

| Frigates | Cost (\$M) | Displacement (t) | Range (NM) | Length (m) |
|------------------------------|------------|------------------|------------|------------|
| Anzac-class frigate | 707.18 | 3,600 | 6,000 | 118.00 |
| Naresuan-class frigate | 123.89 | 2,985 | 4,000 | 120.50 |
| Type 053H2 Jianghu-III class | 138.39 | 1,960 | 4,000 | 103.20 |
| Type 054A Jiangkai-II class | 246.00 | 4,053 | 8,025 | 134.10 |
| Absalon class | 277.71 | 6,600 | 9,000 | 137.00 |
| Iver Huitfeldt class | 393.16 | 6,645 | 9,300 | 138.70 |
| Aquitaine class | 923.38 | 6,000 | 6,000 | 142.00 |
| La Fayette class | 508.54 | 3,800 | 4,000 | 125.00 |
| Brandenburg class | 613.66 | 4,490 | 4,000 | 138.85 |
| F125 Baden-Württemberg class | 1,212.09 | 7,200 | 4,000 | 149.52 |
| Sachsen class | 1,329.45 | 5,800 | 4,000 | 143.00 |
| Valour class | 443.08 | 3,700 | 8,000 | 121.00 |
| Godavari class | 172.39 | 3,850 | 4,500 | 126.40 |
| Project 17A class | 1,029.80 | 6,670 | 1,000 | 149.00 |
| Shivalik class | 694.64 | 6,200 | 5,000 | 142.50 |
| Carlo Bergamini class | 686.52 | 6,700 | 6,000 | 144.60 |
| Maestrale class | 220.34 | 3,040 | 6,000 | 122.70 |
| De Zeven Provinciën class | 827.37 | 6,050 | 4,000 | 144.24 |
| Karel Doorman class | 324.85 | 3,320 | 4,500 | 122.00 |
| Fridtjof Nansen class | 930.14 | 5,290 | 4,500 | 134.00 |
| Zulfiqar class | 983.73 | 3,144 | 4,000 | 123.20 |
| Admiral Gorshkov class | 271.05 | 5,400 | 4,800 | 135.00 |
| Admiral Grigorovich class | 626.66 | 4,000 | 4,850 | 124.80 |
| Gepard class | 386.85 | 1,930 | 4,000 | 102.14 |
| Talwar class | 674.27 | 4,035 | 4,850 | 124.80 |
| Daegu class | 289.99 | 3,592 | 4,500 | 122.00 |
| Incheon class | 274.31 | 3,251 | 4,500 | 114.00 |
| Álvaro de Bazán class F101/4 | 709.43 | 5,800 | 4,500 | 146.70 |
| Álvaro de Bazán class F105 | 1,300.62 | 6,391 | 4,500 | 146.70 |
| Type 22 class Batch 3 | 188.88 | 5,300 | 7,500 | 148.10 |
| Type 23 class | 439.39 | 4,900 | 7,500 | 133.00 |
| Freedom class SSC | 395.76 | 3,500 | 3,500 | 115.00 |
| Legend class cutter | 670.00 | 4,600 | 12,000 | 127.00 |
| Oliver Hazard Perry class | 164.50 | 4,200 | 4,500 | 138.00 |
| Horizon class | 1,940.31 | 7,050 | 6,100 | 152.87 |

Table 1: Frigates Data

The first step was to analyze correlation between variables. Figure 2 shows the correlation matrix.

| | CST | DIS | RNG | LEN |
|-----|------------|-----------|-------------|------------|
| CST | 1.0000000 | 0.5945806 | -0.16838687 | 0.56074101 |
| DIS | 0.5945806 | 1.0000000 | 0.14562317 | 0.91933569 |
| RNG | -0.1683869 | 0.1456232 | 1.00000000 | 0.06115856 |
| LEN | 0.5607410 | 0.9193357 | 0.06115856 | 1.00000000 |

Figure 2: Frigates Correlation Matrix

The coefficient of correlation between displacement and length is higher than 0.7. This number indicates that the variables have a high correlation, and a multicollinearity check will be

needed by using a variance inflation factor (VIF) in case of Multiple Regressions using those two variables.

The next step is to run the regressions and apply the tests. Seven model regressions were generated and tested as listed below:

- Model 1: Multiple regression between Cost (CST), Displacement (DIS), Range (RNG) and Length (LEN);
- Model 2: Multiple regression between Cost (CST), Displacement (DIS) and Range (RNG);
- Model 3: Multiple regression between Cost (CST), Displacement (DIS) and Length (LEN);
- Model 4: Multiple regression between Cost (CST), Range (RNG) and Length (LEN);
- Model 5: Linear regression between Cost (CST) and Length (LEN);
- Model 6: Linear regression between Cost (CST) and Range (RNG);
- Model 7: Linear regression between Cost (CST) and Displacement (DIS).

Below are the outputs for regressions generated in R©. In the case of the frigates, models five and seven pass the regression tests. The slope of the regression line matches what the independent variables are showing; the F-Stat was less than 0.20; and the p-value was also less than 0.20. These two models were kept for further consideration in phase two.

| | Dependent variable: | | | | | | |
|-------------------------|------------------------|-----------------------|-------------------------|---------------------------|----------------------------|-------------------------|-----------------------------|
| | (1) | (2) | (3) | CST (4) | (5) | (6) | (7) |
| DIS | 0.184* (0.099) | 0.176*** (0.038) | 0.142 (0.100) | | | | 0.165*** (0.039) |
| RNG | -0.053* (0.028) | -0.052* (0.027) | | -0.041 (0.028) | | -0.034 (0.034) | |
| LEN | -1.010 (11.079) | | 2.856 (11.299) | 17.947*** (4.451) | 17.557*** (4.513) | | |
| Constant | 149.507 (1,079.250) | 53.231 (220.193) | -442.260 (1,071.107) | -1,542.377** (599.399) | -1,708.965*** (597.248) | 783.732*** (196.550) | -176.046 (192.079) |
| Observations | 35 | 35 | 35 | 35 | 35 | 35 | 35 |
| R ² | 0.420 | 0.420 | 0.355 | 0.356 | 0.314 | 0.028 | 0.354 |
| Adjusted R ² | 0.364 | 0.384 | 0.314 | 0.315 | 0.294 | -0.001 | 0.334 |
| Residual Std. Error | 328.580 (df= 31) | 323.449 (df= 32) | 341.125 (df= 32) | 340.900 (df= 32) | 346.270 (df= 33) | 412.233 (df= 33) | 336.252 (df= 33) |
| F Statistic | 7.486*** (df= 3; 31) | 11.584*** (df= 2; 32) | 8.799*** (df= 2; 32) | 8.832*** (df= 2; 32) | 15.135*** (df= 1; 33) | 0.963 (df= 1; 33) | 18.046*** (df= 1; 33) |
| Note: | | | | | | | *p<0.1; **p<0.05; ***p<0.01 |

Figure 3: Frigates Regression Results

Tests performed until now were part of phase one of the Hierarchy of Regression. In phase two, a comparison was developed between regressions that passed phase one. Table 2 compares models five and seven. It is clear to see that model seven is better than model five, because model seven has a higher R-Square, lower Standard error, and lower CV than model five. Based on the analysis, the best regression to use is the cost and displacement equation:

$$\text{Cost (\$M)} = -176,045 + 0.165287 * \text{displacement (t)}.$$

| Regression | R Square | Std Error | CV |
|----------------------|----------|-----------|-----|
| Cost vs Displacement | 0.353525 | 336.2515 | 56% |
| Cost vs Length | 0.314429 | 346.2698 | 57% |

Table 2: Frigates Regressions Comparison

Since model seven is selected, regression tests between cost and displacement are performed to check the regressions required conditions. The tests are applied to check if the error variable is normal and to check if the error variance is constant (homoskedasticity test).

Figure 4 is R's © output for the residual normality test. The Shapiro-Wilk normality test was used in this instance. The null hypothesis is that the errors follow a normal distribution. The null hypothesis is rejected if $p < 0.05$. Since p-value is greater than 0.05, there is no evidence that the data is drawn from anything other than a normal distribution.

```
> #residual normality test
> #Shapiro-Wilk normality test
      Shapiro-Wilk normality test
data:  residuos
W = 0.97119, p-value = 0.4767

> #p-value < 0.05 there is statistical evidence that leads to the rejection of H0
> #H0: residuals follow normal distribution.
```

Figure 4: Frigates Model 7 Regression – Residual Normality

Figure 5 shows R's © output to check if the error variance is constant (homoscedasticity test). Breusch-Pagan's test was used to verify heteroskedasticity.

```
#heteroscedasticity test
> #studentized Breusch-Pagan test
> bptest(model7)

      studentized Breusch-Pagan test

data:  model7
BP = 4.9348, df = 1, p-value = 0.02632

> #p-value < 0,05 there is statistical evidence that leads to the rejection of H0.
> # Ho: residuals have homoscedasticity
```

Figure 5 Frigates Model 7 - Heteroskedasticity

The null hypothesis is that the variances are equal across the groups. Therefore, if the p-value is under 0.05, the answer is to reject the null hypothesis and conclude the data is not meeting the assumption of homogeneity of variances. In this case, variances were not equal across the groups.

Since the regression did not pass this test, heteroskedasticity-consistent standard errors were used to allow the fitting of a model that does contain heteroskedastic residuals. Figure 6 presents the new model.

```
> #If correction of robust standard errors is necessary
> coefTest(model7, vcov = vcovHC(model7, type="HC0"))

t test of coefficients:

              Estimate Std. Error t value Pr(>|t|)
(Intercept) -176.046049  185.611524  -0.9485 0.3497872
base$DIS      0.165287    0.044352   3.7267 0.0007256 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 6: Frigates Final Regression

Although there were changes in some statistical elements, the regression equation remained the same: $\text{Cost (\$M)} = -176.0461 + 0.1653 * \text{Displacement (t)}$. This equation moves on to the next module to account for uncertainty via a Monte Carlo Simulation. Figure 7 shows the regression chart.

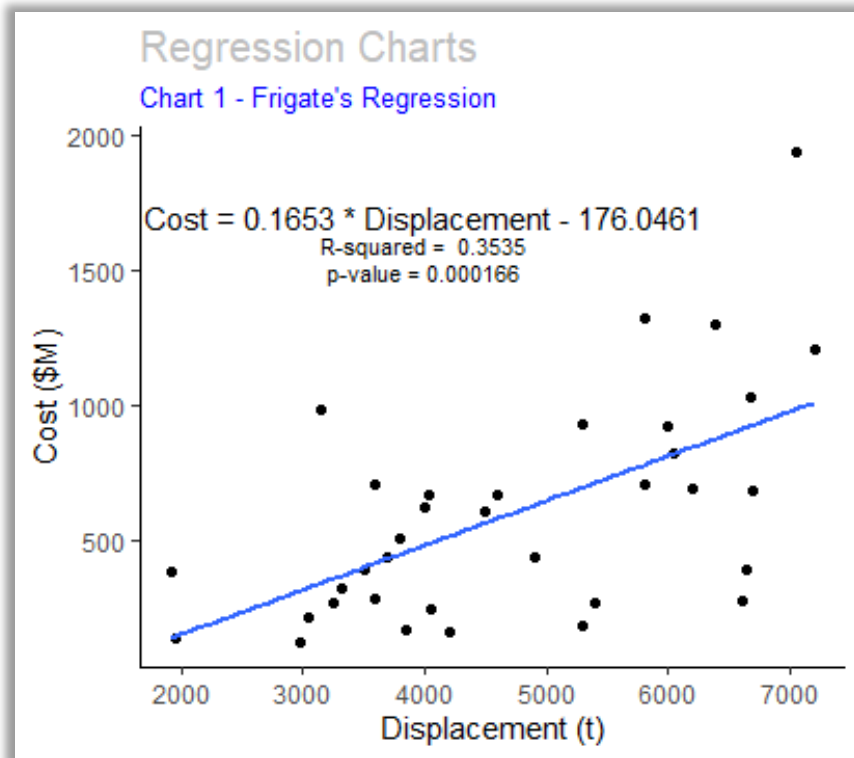


Figure 7: Frigates Regression Chart

2 – PATROLS REGRESSIONS AND COST EQUATIONS

The data for the patrols' regression, consists of cost and technical information like displacement, range, and length. Table 3 shows the data array. The complete list with sources is listed in Appendix D, 2 – Patrols Data. The same regression analysis process was performed on the patrols data to determine the best CER.

| Patrols | Cost (\$M) | Displacement (t) | Range (NM) | Length (m) |
|---|------------|------------------|------------|------------|
| Port of Spain class OPV | 83.40 | 2200 | 5500 | 90.50 |
| Armidale class | 30.80 | 300 | 3000 | 56.80 |
| ASMAR PZM class OPV | 70.80 | 1850 | 8600 | 80.60 |
| Buque de Accion Maritima class OPV | 190.70 | 2860 | 8700 | 93.90 |
| Cape class | 32.90 | 300 | 4000 | 57.80 |
| Carnicobar class patrol boat - Fleet I | 8.20 | 293 | 2000 | 48.90 |
| Carnicobar class patrol boat - Fleet II | 8.20 | 315 | 2000 | 48.90 |
| Classe Yoon Youngha | 44.90 | 570 | 1998 | 63.00 |
| Clyde class OPV | 74.30 | 2000 | 5500 | 81.50 |
| Comandanti-class | 112.70 | 1520 | 3500 | 88.60 |
| Cyclone class patrol boat | 31.70 | 336 | 2500 | 55.00 |
| Flyvefisken class | 31.60 | 450 | 3860 | 54.00 |
| Harry DeWolf class OPV* | 341.70 | 5800 | 6800 | 97.00 |
| Hero class | 20.90 | 257 | 2000 | 42.80 |
| Holland class OPV | 213.80 | 3750 | 5000 | 108.40 |
| Holland-class offshore patrol vessel | 170.80 | 3750 | 5000 | 108.40 |
| Knud Rasmussen class OPV | 105.00 | 2050 | 3000 | 71.80 |
| Meteoro-class offshore patrol vessel | 114.90 | 2860 | 3500 | 93.90 |
| Orca class | 11.00 | 210 | 660 | 33.00 |
| Pacific class | 11.30 | 162 | 2500 | 31.50 |
| Pattani class OPV | 68.80 | 1460 | 3500 | 95.50 |
| Protector class OPV | 75.90 | 1900 | 6000 | 85.00 |
| Rotoiti class IPV | 23.10 | 340 | 3000 | 55.00 |
| Rubin class patrol boat | 40.20 | 630 | 3500 | 62.50 |
| Samuel Beckett class | 108.60 | 2256 | 6000 | 90.00 |
| Sankalp-class | 137.40 | 2020 | 6500 | 105.00 |
| Sentinel class | 70.20 | 359 | 2500 | 46.80 |
| Sirio class OPV | 93.40 | 1549 | 330 | 88.60 |
| Stan Patrol 4207 | 10.20 | 239 | 2450 | 42.80 |
| Svalbard class OPV | 88.60 | 6375 | - | 103.70 |
| Turva class OPV | 151.70 | 4000 | - | 95.90 |
| Viana Do Castelo class OPV | 70.80 | 1600 | - | 83.10 |
| Yoon Youngha-class patrol vessel | 50.20 | 570 | 1998 | 63.00 |

Table 3: Patrols Data

The first step is to analyze correlation between variables. Figure 8 shows the correlation matrix.

| | CST | DIS | RNG | LEN |
|-----|-----------|-----------|-----------|-----------|
| CST | 1.0000000 | 0.9528587 | 0.6003125 | 0.7745443 |
| DIS | 0.9528587 | 1.0000000 | 0.6446105 | 0.8339642 |
| RNG | 0.6003125 | 0.6446105 | 1.0000000 | 0.6372784 |
| LEN | 0.7745443 | 0.8339642 | 0.6372784 | 1.0000000 |

Figure 8: Patrols Correlation Matrix

The coefficient of correlation between displacement and length is higher than 0.7. This indicates that the variables have a high correlation, and a multicollinearity check is needed by using variance inflation factor (VIF) in case of Multiple Regressions using those two variables.

The next step is to run the regressions and apply the tests. Seven model regressions were generated and tested as listed below:

- Model 1: Multiple regression between Cost (CST), Displacement (DIS), Range (RNG) and Length (LEN);
- Model 2: Multiple regression between Cost (CST), Displacement (DIS) and Range (RNG);
- Model 3: Multiple regression between Cost (CST), Displacement (DIS) and Length (LEN);
- Model 4: Multiple regression between Cost (CST), Range (RNG) and Length (LEN);
- Model 5: Linear regression between Cost (CST) and Range (RNG);
- Model 6: Linear regression between Cost (CST) and Length (LEN);
- Model 7: Linear regression between Cost (CST) and Displacement (DIS).

Erro! Fonte de referência não encontrada. Figure 9 lists the outputs for the regressions generated in R©. In the case of patrols, models three, five, six and seven pass the regression tests. The slope of the regression line matches what the independent variables are showing; the F-Stat is less than 0.20; and the p-value is also less than 0.20. These four models are kept for further consideration in phase two.

| | Dependent variable: | | | | | | |
|-------------------------|------------------------|-------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| | (1) | (2) | (3) | CST (4) | (5) | (6) | (7) |
| DIS | 0.055*** (0.006) | 0.052*** (0.004) | 0.026*** (0.008) | | | | 0.036*** (0.005) |
| RNG | -0.0005 (0.003) | -0.001 (0.003) | | 0.006 (0.005) | 0.021*** (0.005) | | |
| LEN | -0.197 (0.352) | | 0.888 (0.540) | 2.108*** (0.492) | | 2.343*** (0.361) | |
| Constant | 16.623 (18.711) | 7.461 (8.968) | -26.322 (30.938) | -95.108*** (28.461) | -1.465 (23.182) | -90.258*** (27.769) | 21.547* (10.689) |
| Observations | 30 | 30 | 33 | 30 | 30 | 33 | 33 |
| R ² | 0.909 | 0.908 | 0.689 | 0.619 | 0.360 | 0.576 | 0.661 |
| Adjusted R ² | 0.899 | 0.901 | 0.669 | 0.591 | 0.338 | 0.562 | 0.650 |
| Residual Std. Error | 23.533 (df = 26) | 23.232 (df = 27) | 41.222 (df = 30) | 47.342 (df = 27) | 60.242 (df = 28) | 47.377 (df = 31) | 42.337 (df = 31) |
| F Statistic | 86.954*** (df = 3; 26) | 133.672*** (df = 2; 27) | 33.276*** (df = 2; 30) | 21.942*** (df = 2; 27) | 15.776*** (df = 1; 28) | 42.095*** (df = 1; 31) | 60.534*** (df = 1; 31) |

Note: *p<0.1; **p<0.05; ***p<0.01

Figure 9: Patrols Regressions Results

Tests performed until now were part of phase one of the Hierarchy of Regression. Table 4 compares models three, five, six and seven. Model three is the best choice. It has a higher R-Square, lower standard error, and lower CV. Based on the analysis, the best regression to use is the cost, displacement, and length equation:

$$\text{Cost (\$M)} = -26.3216 + 0.0256791 * \text{displacement (t)} + 0.887922 * \text{length (m)}.$$

| Regression | R Square | Std Error | CV |
|---------------------------------|----------|-----------|-----|
| Cost vs Displacement and Length | 0.668574 | 41.22203 | 50% |
| Cost vs Range | 0.360375 | 60.24218 | 76% |
| Cost vs Length | 0.575893 | 47.37703 | 58% |
| Cost vs Displacement | 0.661330 | 42.33687 | 52% |

Table 4: Patrol Regressions Comparison

Since model three passed the Hierarchy of Regression, additional tests were applied to check if the error variable is normal and to check if the error variance is constant (homoskedasticity test). Figure 10 shows R's © output for the residual normality test. The null hypothesis is that the errors follow a normal distribution. The null hypothesis is rejected if $p < 0.05$. Since the p-value is less than 0.05, there is evidence that the data was not drawn from a normal distribution. However, when the sample size is sufficiently large (>30), the normality assumption is not needed as the Central Limit Theorem ensures that the distribution of residuals will approximate normality.

```
> #residual normality test
> #Shapiro-Wilk normality test

      Shapiro-Wilk normality test

data:  residuos
W = 0.7814, p-value = 1.454e-05

> #p-value < 0.05 there is statistical evidence that leads to the rejection of H0
> #H0: residuals follow normal distribution.
> #However, by the central limit theorem, if n >= 30, one can relax with the assumption
of normality of the residuals.
```

Figure 10: Patrols Model 3 Regression – Residual Normality

Figure 11 shows R's © output to check if the error variance is constant (homoskedasticity test). Breusch-Pagan's test was used in this case.

```
> #heteroscedasticity test
> #studentized Breusch-Pagan test
> bptest(model3)

      studentized Breusch-Pagan test

data:  model3
BP = 24.415, df = 2, p-value = 4.994e-06

> #p-value < 0,05 there is statistical evidence that leads to the rejection of H0.
> # Ho: residuals have homoscedasticity
```

Figure 11: Patrols Model 3 Regression - Heteroskedasticity

The null hypothesis is that the variances are equal across the groups. Therefore, if the p-value is under 0.05, the answer is to reject the null hypothesis and conclude that the data is not meeting the assumption of homogeneity of variances. In this instance, variances are not equal across the groups.

Since the regression did not pass this test, heteroskedasticity-consistent standard errors were used to fit into a model that does contain heteroskedastic residuals. The new model is detailed in Figure 12.

```
> #If correction of robust standard errors is necessary
> coeftest(model3, vcov = vcovHC(model3, type="HC0"))

t test of coefficients:

              Estimate Std. Error t value Pr(>|t|)
(Intercept) -26.321647  35.527315  -0.7409   0.4645
base$DIS     0.025679   0.020011   1.2832   0.2092
base$LEN     0.887922   0.853654   1.0401   0.3066
```

Figure 12: Patrols Final Regression

The regression equation remained the same: $\text{Cost (\$M)} = -26.3216 + 0.0257 * \text{Displacement (t)} + 0.8879 * \text{Length (m)}$. This equation moves on to run through Monte Carlo Simulations, which will account for uncertainty within the equation. This regression did not result in a scatterplot due to multiple variables (i.e., Multiple Regression).

The correlation between displacement and length is higher than 0.7. Therefore, a multicollinearity check is needed by using a variance inflation factor (VIF). Figure 13 shows the output for the multicollinearity test, since VIF is less than 10, there are no problems of multicollinearity (Hair 2014).

```
vif(model3)
base$DIS base$LEN
2.958595 2.958595
> #VIF > 10.0 may indicate problems of multicollinearity
```

Figure 13: Patrols VIF Test Results

3 – SUBMARINES REGRESSIONS AND COST EQUATIONS

The data for the submarines’ regression, consists of cost and technical information like displacement, range, and length. Table 5 shows the data array. The complete list with sources is listed in Appendix D, 3 – Submarines Data. This data was also run through the regression steps like the previous data. After this data is analyzed, the next step is to run Monte Carlo Simulations.

| Submarines | Cost (\$M) | Length (m) | Beam (m) | Displacement (t) |
|---------------------------------|------------|------------|----------|------------------|
| Collins class SSK | 1,299.14 | 77.42 | 7.80 | 3,100 |
| Type 035 Ming class SSK | 113.87 | 76.00 | 7.60 | 1,584 |
| Type 041 Yuan class SSK | 683.29 | 77.60 | 8.40 | 2,700 |
| Agosta 90B class SSK | 845.59 | 76.20 | 6.80 | 1,570 |
| Scorpène CA-2000 class SSK | 590.24 | 61.70 | 6.20 | 0 |
| Dolphin 2 class SSK | 753.48 | 68.60 | 6.80 | 2,050 |
| Shishumar class (Type 209/1500) | 872.18 | 64.40 | 6.50 | 1,660 |
| Heroine class (Type 209/1400) | 362.58 | 62.00 | 7.60 | 1,545 |
| Type 210 'Ula' class SSK | 147.50 | 59.00 | 5.40 | 1,040 |
| Type 212A class SSK | 655.82 | 56.00 | 6.80 | 1,524 |
| Type 214 class SSK | 399.21 | 65.00 | 6.30 | 1,690 |
| Type 216 class SSK | 1,770.41 | 90.00 | 8.10 | 4,000 |
| Sōryū class | 602.11 | 84.00 | 9.10 | 2,900 |
| Walrus class | 467.63 | 67.73 | 8.40 | 2,465 |
| Kilo class | 347.27 | 73.80 | 9.90 | 2,350 |
| Jang Bogo class | 413.45 | 61.00 | 6.30 | 1,180 |
| S-90 class | 1,180.63 | 81.05 | 11.68 | 3,200 |
| Blekinge class | 524.22 | 63.00 | 6.40 | 1,800 |

Table 5: Submarines Data

The first step is to analyze correlation between variables. Figure 14 shows the correlation matrix.

| | CST | LEN | BEAM | DIS. SUR |
|----------|-----------|-----------|-----------|-----------|
| CST | 1.0000000 | 0.6100425 | 0.3155114 | 0.7716423 |
| LEN | 0.6100425 | 1.0000000 | 0.6455690 | 0.8425820 |
| BEAM | 0.3155114 | 0.6455690 | 1.0000000 | 0.6922294 |
| DIS. SUR | 0.7716423 | 0.8425820 | 0.6922294 | 1.0000000 |

Figure 14: Submarines Correlation Matrix

The coefficient of correlation between displacement and length is higher than 0.7. This indicates the variables have a high correlation, and a multicollinearity check is needed by using a variance inflation factor (VIF) in case of Multiple Regressions.

Next step is to run the regressions and apply the tests. Seven model regressions were generated and tested as listed below:

- Model 1: Multiple regression between Cost (CST), Displacement (DIS.SUR), Beam (BEAM) and Length (LEN);
- Model 2: Multiple regression between Cost (CST), Beam (BEAM) and Length (LEN);

- Model 3: Multiple regression between Cost (CST), Displacement (DIS.SUR) and Length (LEN);
- Model 4: Multiple regression between Cost (CST), Displacement (DIS.SUR) and Beam (BEAM);
- Model 5: Linear regression between Cost (CST) and Length (LEN);
- Model 6: Linear regression between Cost (CST) and Beam (BEAM);
- Model 7: Linear regression between Cost (CST) and Displacement (DIS.SUR).

Figure 15 lists the outputs for regressions generated in R[©]. In the case of the submarines, models five, six, and seven pass the regression tests. The slope of the regression line matches what the independent variables show; the F-Stat is less than 0.20; and the p-value is also less than 0.20. These three models were kept for further consideration in phase two.

| | Dependent variable: | | | | | | |
|-------------------------|-----------------------|--------------------------|------------------------|------------------------|--------------------------|---------------------|------------------------|
| | (1) | (2) | (3) | CST (4) | (5) | (6) | (7) |
| LEN | -2.198 (12.955) | 30.483** (11.802) | -6.150 (13.933) | | 26.225*** (8.640) | | |
| BEAM | -114.215* (60.023) | -39.902 (73.315) | | -115.847* (57.155) | | 85.630 (63.882) | |
| DIS.SUR | 0.579*** (0.162) | | 0.467** (0.165) | 0.559*** (0.109) | | | 0.406*** (0.086) |
| Constant | 463.670 (684.717) | -1,171.499* (626.182) | 108.724 (717.842) | 362.913 (328.572) | -1,174.025* (612.238) | 20.893 (492.354) | -195.240 (196.948) |
| Observations | 17 | 18 | 17 | 17 | 18 | 18 | 17 |
| R ² | 0.688 | 0.378 | 0.601 | 0.687 | 0.365 | 0.101 | 0.595 |
| Adjusted R ² | 0.616 | 0.295 | 0.544 | 0.643 | 0.326 | 0.045 | 0.568 |
| Residual Std. Error | 265.367 (df = 13) | 349.223 (df = 15) | 289.141 (df = 14) | 255.997 (df = 14) | 341.456 (df = 16) | 406.428 (df = 16) | 281.273 (df = 15) |
| F Statistic | 9.552*** (df = 3; 13) | 4.552** (df = 2; 15) | 10.543*** (df = 2; 14) | 15.380*** (df = 2; 14) | 9.214*** (df = 1; 16) | 1.797 (df = 1; 16) | 22.077*** (df = 1; 15) |

Note: *p<0.1; **p<0.05; ***p<0.01

Figure 15: Submarines Regressions Results

Tests performed until now were part of phase one of the Hierarchy of Regression. In phase two, a comparison is developed between regressions that passed phase one. Table 6 compares models five, six, and seven. Clearly, model seven is the best one as it has the higher R-Square, lower standard error, and lower CV. The regression selected is cost versus displacement, which yields the equation: $Cost (\$M) = -195.239 + 0.4059 * Displacement (t)$.

| Regression | R Square | Std Error | CV |
|----------------------|----------|-----------|-----|
| Cost vs Length | 0.365428 | 341.4559 | 51% |
| Cost vs Beam | 0.100963 | 406.4276 | 61% |
| Cost vs Displacement | 0.595432 | 281.2729 | 42% |

Table 6: Submarines Regression Comparison

Since model seven was chosen, regression tests between cost and displacement are performed to check the regressions required conditions. The tests are applied to check if the error variable is normal and to check if the error variance is constant (homoskedasticity test).

Figure 16 is R's © output for the residual normality test. It used the Shapiro-Wilk normality test. The null hypothesis is that the errors follow a normal distribution. The null hypothesis is rejected if $p < 0.05$. Since p-value is greater than 0.05, there is no evidence that the data is drawn from anything other than a normal distribution.

```
> #residual normality test
> #Shapiro-Wilk normality test
      Shapiro-Wilk normality test
data:  residuos
W = 0.94036, p-value = 0.3225

> #p-value < 0.05 there is statistical evidence that leads to the rejection of H0
> #H0: residuals follow normal distribution.
```

Figure 16: Model 7 Regression – Residual Normality

Figure 17 shows R's © output to check if the error variance is constant (homoscedasticity test). Breusch-Pagan's test is used in this case.

```
> #heteroscedasticity test
> #studentized Breusch-Pagan test
> bptest(model7)

      studentized Breusch-Pagan test

data:  model7
BP = 0.9284, df = 1, p-value = 0.3353

> #p-value < 0,05 there is statistical evidence that leads to the rejection of H0.
> # Ho: residuals have homoscedasticity
```

Figure 17: Submarines Model 7 - Heteroskedasticity

The null hypothesis is that the variances are equal across the groups. Therefore, if the p-value is under 0.05, the answer is to reject the null hypothesis and conclude the data is not meeting the assumption of homogeneity of variances. In this case, variances are equal across the groups. Since the regression passed in all tests, the final regression equation is:

$\text{Cost (\$M)} = -195.239 + 0.4059 * \text{Displacement (t)}$. Figure 18 shows the regression chart.

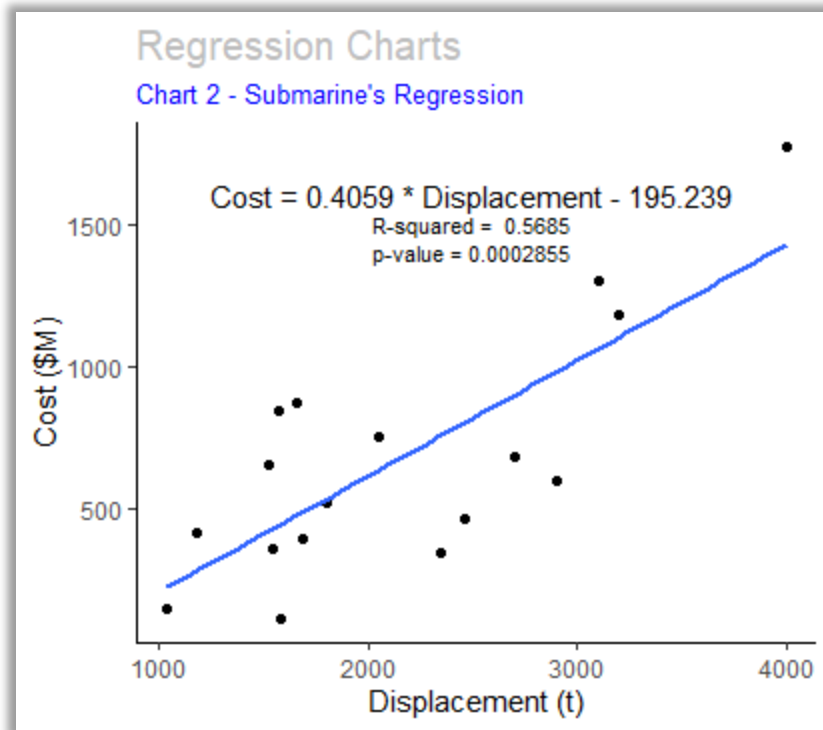


Figure 18: Submarines Regression Chart

D – MODULE TWO: MONTE CARLO SIMULATION

The result of a CER must be handled as a probability distribution and not as a deterministic number due to the uncertainty. The objective for Module two is to obtain a probability distribution function for the cost of each system, accounting for cost equations and input variable uncertainties. Each of the cost equations obtained in the regression module will be the input for the simulation module.

The R© created programming script emulates how the software @RISK handles Monte Carlo Simulations and is available in Appendix B: Monte Carlo Simulation Programming Script. The simulations in each software program yield significantly similar results. This can be seen in Figure 21, Figure 24, and Figure 27, which show the comparisons side-by-side. Since it is a simulation, it is impossible to get the exact same result. Considering that @RISK license is expensive, being able to provide a script for Monte Carlo Simulation in an open-source software package with similar results is a major accomplishment.

In the cases of the frigates and submarines, the independent variable is displacement. In the case of patrols, there are two independent variables – displacement and length. The simulations for the independent variables are set to 10,000 iterations as is normal practice for DoD cost estimating. SMEs from the Brazilian Navy suggest choosing a triangular distribution for

the independent variable by assigning a minimum, most likely, and maximum value for each independent variable. Figure 19, Figure 22, and Figure 25 show the results of the triangular values based on the SME assessment. The patrol simulation accounts for the correlation between displacement and length.

The cost distribution is considered lognormal. Lognormal distributions have a defined lower bound that is never less than zero and have an upper bound of infinity, thus providing at least some probability of a large cost overrun. In the absence of better information, choose lognormal distribution as the shape of the uncertainty distribution (Thomas and Fitch 2014).

The amount of budget set for each system depends on the amount of risk the decision-maker is willing to take. The eightieth percentile is used as a demonstrative example and will be considered in module 3, the NLP model. The distribution charts that result from the simulation charts are produced by R© for each vessel type as depicted in Figure 20, Figure 23, and Figure 26 in the following pages.

1 – FRIGATES COST DISTRIBUTION

| Displacement | t |
|--------------------------|-------------------|
| Minimum | 1,960.0000 |
| Most Likely | 2,150.0000 |
| Maximun | 2,350.0000 |
| Mean (Calculated) | 2,152.2410 |

Figure 19: SME Assessment for Frigates Displacement

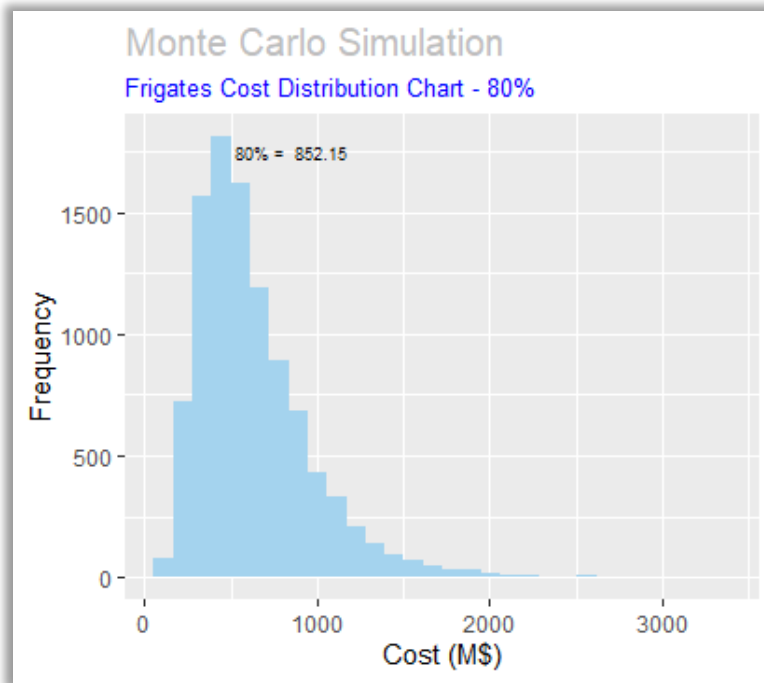


Figure 20: Frigates Cost Cumulative Distribution Function

| Statistics | M (\$) - R | M (\$) - @ RISK |
|--------------------|-----------------|-----------------|
| Mean | 634.4595 | 636.1300 |
| Standard Deviation | 336.3000 | 336.7600 |
| 80% | 852.1504 | 854.2200 |

Figure 21: Frigates Cost Distribution Function – Main Parameters and Comparison

2 – PATROLS COST DISTRIBUTION

| | Displacement (t) | Lenght (m) |
|--------------------------|-------------------|----------------|
| Minimum | 1,250.0000 | 68.0000 |
| Most Likely | 1,500.0000 | 70.0000 |
| Maximun | 1,780.0000 | 74.0000 |
| Mean (Calculated) | 1,508.4830 | 70.6463 |

Figure 22: SME Assessment for Patrols Displacement and Length

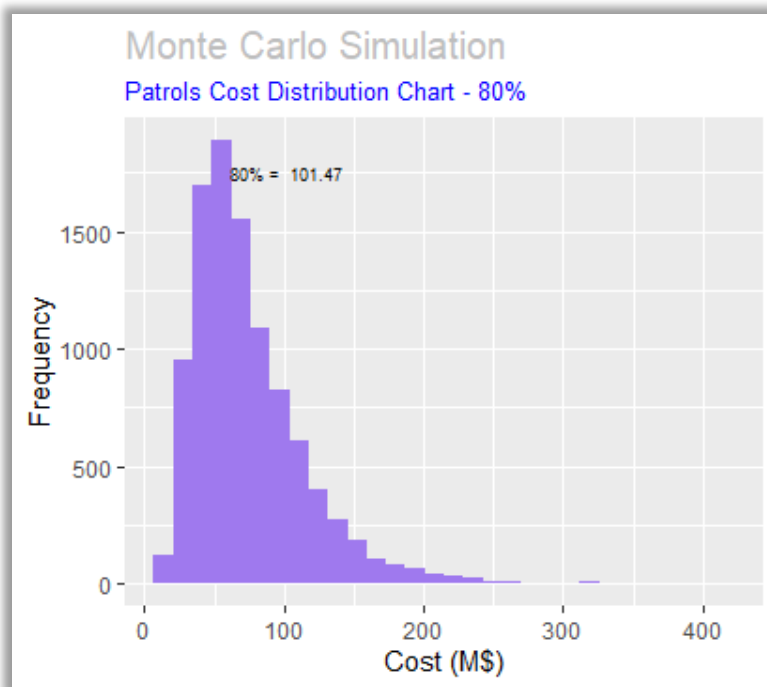


Figure 23: Patrols Cost Cumulative Distribution Function

| Statistics | M (\$) - R | M (\$) - @ RISK |
|--------------------|-----------------|-----------------|
| Mean | 75.1623 | 75.2100 |
| Standard Deviation | 41.2220 | 41.4700 |
| 80% | 101.4689 | 101.7300 |

Figure 24: Patrols Cost Distribution Function – Main Parameters and Comparison

3 – SUBMARINES COST DISTRIBUTION

| Displacement | t |
|--------------------------|-------------------|
| Minimum | 1,960.0000 |
| Most Likely | 2,150.0000 |
| Maximun | 2,350.0000 |
| Mean (Calculated) | 2,152.2410 |

Figure 25: SME Assessment for Submarine Displacement

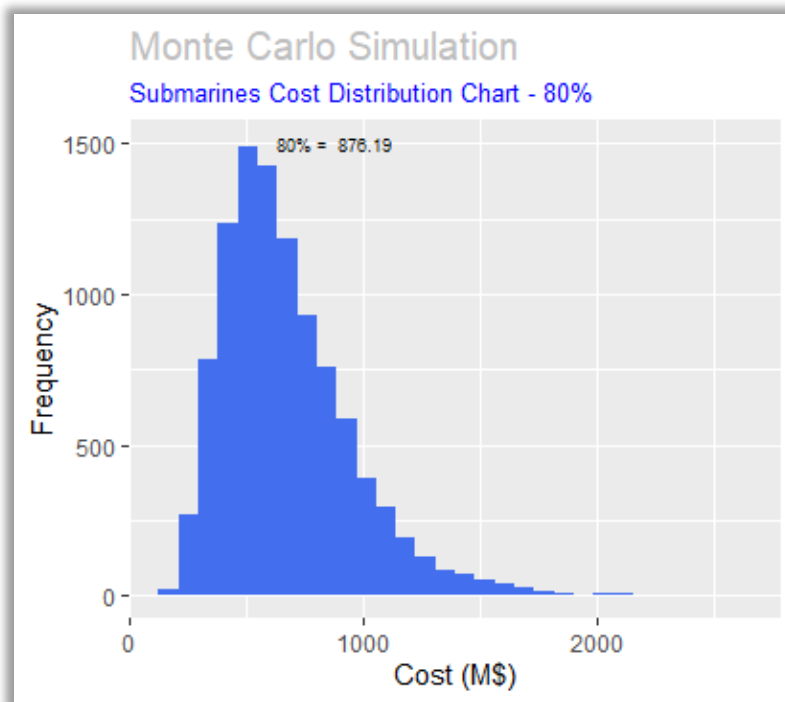


Figure 26: Submarines Cost Cumulative Distribution Function

| Statistics | M (\$) - R | M (\$) - @ RISK |
|--------------------|-----------------|-----------------|
| Mean | 678.3546 | 678.8100 |
| Standard Deviation | 281.3000 | 283.2200 |
| 80% | 876.1922 | 877.7700 |

Figure 27: Submarines Cost Distribution Function – Main Parameters and Comparison

E – MODULE THREE: NON-LINEAR PROGRAMMING MODEL

1 - THEORETICAL NON-LINEAR PROGRAMMING MODEL

The model created is based on a theoretical non-linear programming model description. The non-linear programming structure is presented in the Managerial Decision Modeling textbook by Balakrishnan, Render, Stair, and Munson. The textbook discusses when it is appropriate to use the NLP model and how to apply decision variables, objective functions, and constraints (Balakrishnan, et al. 2017). NLP is selected to comply with sponsor's request to use linear or goal programming. After running regressions and simulations, NLP completes the model by determining the best combination of the three decision variables within this study. Below is the mathematical notation for the NLP model:

Decision Variables:

Q_F = Number of frigates

Q_P = Number of patrol ships

Q_S = Number of submarines

$$\text{Objective Function: Min Portfolio cost} = \frac{\sum_{i=0}^{Q_F} C_{Fi}}{Q_F} \cdot Q_F + \frac{\sum_{j=0}^{Q_P} C_{Pj}}{Q_P} \cdot Q_P + \frac{\sum_{k=0}^{Q_S} C_{Sk}}{Q_S} \cdot Q_S$$

Each first unit cost will be determined by the Cumulative Distribution Function from the Monte Carlo Simulation. A learning curve will be applied to determine the cost of each subsequent unit.

Where:

C_F = Frigate cost (Cumulative Distribution Function)

C_P = Cost of the patrol ship (Cumulative Distribution Function)

C_S = Submarine cost (Cumulative Distribution Function)

Constraints (Subject to):

$$Q_F + Q_P + Q_S \geq 16$$

$$Q_P \geq 4Q_S$$

$$Q_F \geq 2Q_S$$

$$Q_P \geq 2Q_F$$

$$Q_F, Q_P, Q_S \geq 1$$

2 – ACCOUNTING FOR LEARNING CURVES WITH A MULTI-VARIABLE MODEL

Learning Curve Theory suggests that as a task is repeated, the performer gets better at it. In manufacturing, it has been found that production improves at a constant rate (Mislick and Nussbaum 2015). Cost equations produced in a reliable cost estimate must account for uncertainty and learning curves. Doing so refines the data analysis to produce a credible cost estimate.

This model uses learning curves, or the assumption that the construction of each subsequent ship will take less time than those that have already been built. Thereby producing a cost savings with each ship procured. The unit theory of learning was used for this model for simplicity's sake. Whether Unit Theory or Cumulative Average Theory is used, the analyst must use only one to maintain consistent unit costs. In Unit Theory, it is assumed that every time the number of units produced is doubled, then a consistent efficiency is gained. For example, when using a learning curve of 90% it is assumed the second unit will take 90% as long to produce as the first one. Similarly, the fourth unit will take 90% as long to produce as the second one. The pattern is assumed to continue ad infinitum.

Typically, a Linear Programming model has fixed average unit costs, where each unit has the same cost as the one before it. When using learning curves, the cost of each unit is known, but the average unit cost changes depending on the amount of each unit purchased. A simplified version of a cost equation is the average unit cost multiplied by the number of units purchased.

For this capstone project, the objective function changed from a single unit cost to an average unit cost (i.e., total cost of units divided by units purchased) multiplied by the number of units purchased. The units purchased part of the formula cancels out, which yields total cost for that unit. The analyst can then create a table with total costs based on the number of units purchased. Using this method simplifies the calculations and becomes increasingly important as the model allows for larger purchase quantities and more unit types (variables).

Dealing with a changing average unit cost requires a different approach to the cost equation, though. The typical equation to calculate the cost of an individual unit, denoted as x , using learning curves is $Cost = Ax^b$, where A is the theoretical first unit's cost, also known as T1, and b is the learning coefficient. The learning coefficient is calculated by the equation $b = \ln(\text{learning curve})/\ln(2)$. The theoretical first unit cost is an output of the Monte Carlo Simulations discussed previously. This model uses naval military standard learning curves, which normally range from 80% to 85% for shipbuilding (Ghanmi 2017). For this model, three different learning curves are used for each type of vessel included in the theoretical NLP. The slopes of the learning curves and the cost of unit one for each system are set as shown in Table 7. The learning curves are for demonstration purposes only. The patrols learning curve is outside the industry range as

an assumption is made that patrols are produced in significantly higher amounts than frigates or submarines. Thereby, the learning curve has leveled out to 88%.

| | Frigates | Patrols | Submarines |
|-----------------|----------|---------|------------|
| T1 (\$M) | 852.15 | 101.47 | 876.19 |
| Learning Curve* | 80% | 88% | 85% |

* Industry Standard

Table 7: Values and learning Curves used in NLP Model

3 – TRANSLATING THE THEORETICAL NLP MODEL INTO R©

As previously discussed in the Analytical Tool section, R© is an integrated suite of software facilities used for statistical analysis. The following is a description of how R© processes the information input by the NLP model to produce an optimal solution. This step is done after R© has processed the regressions and Monte Carlo Simulations.

The model begins by creating a variable for each of the three decision variables and allows these variables to take an integer value anywhere from 1 to 100. This accounts for all possible portfolio configurations for the model to consider. Each combination, one million in total, is put into a vector. In R©, a vector is a variable that contains multiple pieces of information. In this case, vectors contain the quantity of each ship type that could be bought in the proposed portfolio configuration.

Then, the model used nested ‘if’ loops to combine the systems quantities into vectors and check if each vector created contained a mix of ships that satisfy each of the constraints that have been set. A table was created to store the vectors that were suitable, but not necessarily the optimal solution. Three vectors were created containing the unit costs for each system. To do this, the model uses the first unit costs calculated in the Monte Carlo Simulation and applies their respective learning curves. The model used nested ‘if’ loops to create cumulative cost vectors. The cumulative cost vectors were used to add a column in the table with the possible cost solutions. From this point the model found the solution with the minimum cost for the overall portfolio. The programming script for this process is available in Appendix C, ‘1 Minimum Budget’.

The optimal portfolio solution calculated in this scenario results in 15 frigates, 30 patrols and 1 submarine. The separate costs for frigates, patrols and submarines are \$7252.26M, \$1,948.19M, \$ 876.19M, respectively. Total portfolio cost is \$10,076.64M.

4 – SPONSOR-SUGGESTED ALTERATION

While the initial model itself is a useful tool, the team sponsor submitted additional input to make it more useful in real world situations. The first suggestion was to upgrade the model to make the calculations based on a fixed budget where the target is to maximize acquisitions while simultaneously minimizing the remaining budget. A simple calculation of the fixed budget minus the portfolio cost was added to the NLP objective function. A fixed budget of \$32,030M was developed to test the new model, and therefore, is for demonstration purposes only. This change requires a slightly different programming script, which is available in Appendix C, '2 Fixed Budget'. The changes in the objective function are shown below:

Objective Function: Minimize remaining budget = $32,030 - \frac{\sum_{i=0}^{Q_F} C_{Fi}}{Q_F} \cdot Q_F + \frac{\sum_{j=0}^{Q_P} C_{Pj}}{Q_P} \cdot Q_P + \frac{\sum_{k=0}^{Q_S} C_{Sk}}{Q_S} \cdot Q_S$

The second suggestion was to figure out the shadow price of each type of ship. The shadow price is simply the dollar amount needed to purchase one additional ship. Since the model accounts for three different ship types, the programming calculated three respective shadow prices.

Calculating the shadow prices is a simple process. The model subtracts the total cost of each type of ship from the total cost of buying an additional ship. Shadow prices are easily calculated, but they provide leadership key information. When compared to how much budget remains after purchasing the portfolio, leadership can see how much additional funding is required to procure one more ship.

The optimal portfolio solution calculated in this scenario results in 44 frigates, 95 patrols and 21 submarines. The total costs for frigates, patrols and submarines are \$15,670.98M, \$5,053.03M, and \$ 11,295.90M, respectively. Total sum of the portfolio cost is \$32,019.19M, leaving about \$10.8M or less than 4% of the portfolio unspent. Shadow prices for each system are \$250.21M for frigates, \$43.73M for patrols, and \$424.47M for submarines.

The model makes use of this easily accessible information and its usefulness by producing three charts for decision makers. Figure 28 shows the total cost of each ship type and their respective shadow price. The chart shows decision makers a breakout of costs by system, which gives them a perspective on the ratio of price to shadow price. Since the shadow price for patrols is significantly low compared to the values of the other systems, it is not visible in the chart. Figure 29 shows the shadow price relative to the remaining budget. In this chart, it is easier to compare the remaining budget to the additional funding required for one additional system unit. It would be relatively easier to obtain additional funding for one more patrol than a submarine due to the price difference. Finally, Figure 30 shows the percentage of an additional ship that the remaining budget can purchase. In essence, with the remaining budget left, the program office could buy 23% of a patrol boat. Again, the patrols stand out due to their low price in

comparison to a frigate or submarine. This is just one of the many functionalities R© can produce to analyze cost estimates.

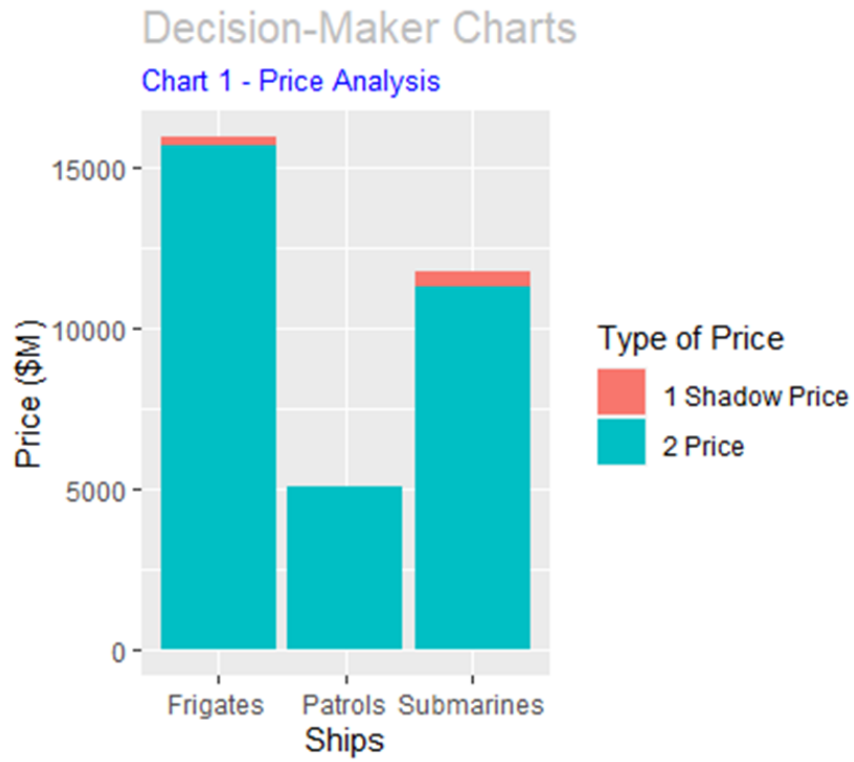


Figure 28: Total Cost of Each Ship and Their Respective Shadow Price

Decision-Maker Charts

Chart 2 - Shadow Price x Remaining Budget

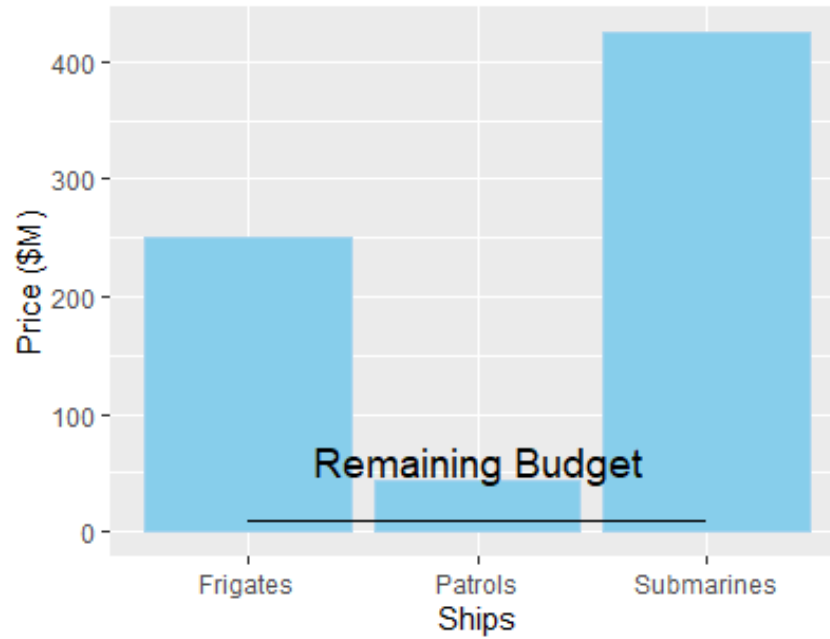


Figure 29: Shadow Prices Relative to the Remaining Budget

Decision-Maker Charts

Chart 3 - Remaining Budget / Shadow Price

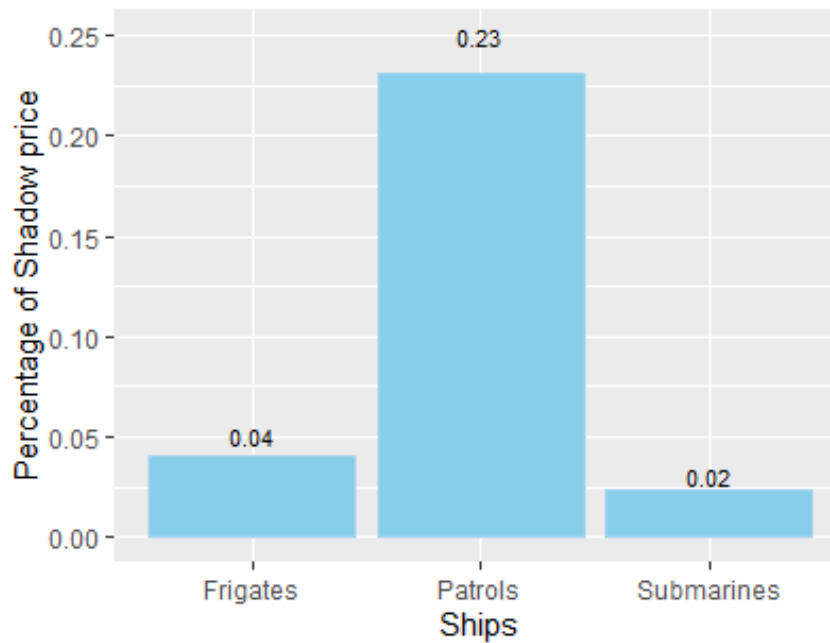


Figure 30: Percentage of Each Ship Type Purchasable with Remaining Budget

V – CONCLUSION

The team learned during this project that it is possible to use different tools and create a new way to analyze data and portfolios. Concepts like linear regression, learning curves, Monte Carlo Simulation, and linear programming are separate tools in cost estimating. The team managed to use them in an integrated way, providing the sponsor with a solution to a problem that at first glance seemed too big for a Capstone Project. Granted, the current process and model is simple, but it still proves there is a different way to look at cost estimating analysis. The model is a good starting point for cost estimators to use for their own portfolios. In this project, notional data was used to test the model in its initial stage. However, real data can be used to follow the same process. The coding scripts are included in Appendix A, B, and C to help users navigate R©. The team foresees additional variables and constraints added to make the model more realistic. Currently, the model only accounts for production costs. SERs, manpower costs, or other restrictions can be added. The model was set up this way to test if a simplified version would work and the team has proven it does. Now that the model has been proven, it is possible to upgrade and refine it. This team's study successfully answered the sponsor's problem.

The analytical tool, R©, proved to be extremely robust for use in data analysis and processing. It stands out not only for its ability to handle large amounts of data, but also for its flexibility in handling different types of needs such as linear regressions and NLP. There were countless hours spent studying and learning the programming language, but as the work progressed, it becomes easier to deal with more complex issues such as testing regressions, Monte Carlo Simulation or creating an NLP model. Developing the graphics is an additional challenge. Accustomed to ready-made graphics packages, such as Excel, it seemed impossible for the team to create graphics that stood out. However, the tool offers an infinite range of graphical resources that can be explored by cost estimators to facilitate the exposure of data to decision makers. Cost estimators can use this model today with the data from their own portfolio. They can give decision makers a more in-depth analysis on how best to spend scarce resources. The final model is ready for someone else to take it a step further to obtain a real-world optimal solution.

Appendix A: Regressions Programming Script

1 Frigates Regression

```
#Calling required packages
library("dplyr") #package to clean the bases
library("ggplot2") #package for plotting graphics
library("lmtest") #package to perform multiple regression tests
library("sandwich") #package to calculate robust standard errors
library("readxl")
library("stargazer")

#csv file location for import
setwd("C:\\Users\\pulch\\Desktop\\Bases Reg\\Reg Frig")
getwd()

#obtaining basic statistical data from the imported file
base <- read.csv2(file = "Data Reg Frig.csv", header = TRUE, dec = ",", sep = ";")
summary(base)

#correlation analysis
cor(na.omit(base))

#running regression for 1st model
model1<-lm(base$CST ~ base$DIS + base$RNG + base$LEN, data= base)
summary(model1)
confint(model1)

#running regression for 2nd model
model2<-lm(base$CST ~ base$DIS + base$RNG, data= base)
summary(model2)
confint(model2)

#running regression for 3rd model
model3<-lm(base$CST ~ base$DIS + base$LEN, data= base)
summary(model3)
confint(model3)

#running regression for 4th model
model4<-lm(base$CST ~ base$LEN + base$RNG, data= base)
summary(model4)
confint(model4)

#running regression for 5th model
model5<-lm(base$CST ~ base$LEN, data= base)
summary(model5)
confint(model5)
```

```

#running regression for 6th model
model6<-lm(base$CST ~ base$RNG, data= base)
summary(model6)
confint(model6)

#running regression for 7th model
model7<-lm(base$CST ~ base$DIS, data= base)
summary(model7)
confint(model7)

#generating summary report
stargazer(model1, model2, model3, model4, model5, model6, model7, out = "Resultado Reg Frig.html")

#choose model
#model 7

#regression graphic
#Creating data
df = data.frame(Displacement = base$DIS, Cost = base$CST)

grp <- ggplot(data = df,
              mapping = aes(x = Displacement, y = Cost)) +
  geom_point()+
  geom_smooth(method="lm", se=0) +
  labs(title="Regression Charts",
       subtitle="Chart 1 - Frigate's Regression") +
  ylab ("Cost ($M)") +
  xlab ("Displacement (t)") +
  theme_classic() +
  annotate("text",x=4000,y=1500,label = "p-value = 0.000166", size = 3) +
  annotate("text",x=4000,y=1600,label = "R-squared = 0.3535", size = 3) +
  annotate("text",x=4000,y=1700,label = "Cost = 0.1653 * Displacement - 176.0461")
grp

grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray") ,
        plot.subtitle = element_text(size = 10, color = "blue"))
grp

#residual normality test
#Shapiro-Wilk normality test
residuos<- model7$residuals
shapiro.test(residuos)
#p-value < 0.05 there is statistical evidence that leads to the rejection of H0
#H0: residuals follow normal distribution.
#However, by the central limit theorem, if n >= 30, one can relax with the assumption of normality of the residuals.

```

```

#heteroscedasticity test
#studentized Breusch-Pagan test
bptest(model7)
#p-value < 0,05 there is statistical evidence that leads to the rejection of H0.
# Ho: residuals have homoscedasticity

#If correction of robust standard errors is necessary
coefest(model7, vcov = vcovHC(model7, type="HC0"))

```

2 Patrols Regression

```

#Calling required packages
library("dplyr") #package to clean the bases
library("ggplot2") #package for plotting graphics
library("lmtest") #package to perform multiple regression tests
library("sandwich") #package to calculate robust standard errors
library("readxl")
library("stargazer")

#csv file location for import
setwd("C:\\Users\\pulch\\Desktop\\Bases Reg\\Reg Patr")
getwd()

#obtaining basic statistical data from the imported file
base <- read.csv2(file = "Data Reg Patr.csv", header = TRUE, dec = ",", sep = ";")
summary(base)

#correlation analysis
cor(na.omit(base))

#running regression for 1st model
model1 <- lm(base$CST ~ base$DIS + base$RNG + base$LEN, data= base)
summary(model1)
confint(model1)

#running regression for 2nd model
model2 <- lm(base$CST ~ base$DIS + base$RNG, data= base)
summary(model2)
confint(model2)

#running regression for 3rd model
model3 <- lm(base$CST ~ base$DIS + base$LEN, data= base)
summary(model3)
confint(model3)

#running regression for 4th model
model4 <- lm(base$CST ~ base$LEN + base$RNG, data= base)

```

```

summary(model4)
confint(model4)

#running regression for 5th model
model5<-lm(base$CST ~ base$RNG, data= base)
summary(model5)
confint(model5)

#running regression for 6th model
model6<-lm(base$CST ~ base$LEN, data= base)
summary(model6)
confint(model6)

#running regression for 7th model
model7<-lm(base$CST ~ base$DIS, data= base)
summary(model7)
confint(model7)

#generating summary report
stargazer(model1, model2, model3, model4, model5, model6, model7, out = "Patr Reg Result.html")

#choose model
#model 3

#residual normality test
#Shapiro-Wilk normality test
residuos<- model3$residuals
shapiro.test(residuos)
#p-value < 0.05 there is statistical evidence that leads to the rejection of H0
#H0: residuals follow normal distribution.
#However, by the central limit theorem, if n >= 30, one can relax with the assumption of normality of the residuals.

#heteroscedasticity test
#studentized Breusch-Pagan test
bptest(model3)
#p-value < 0,05 there is statistical evidence that leads to the rejection of H0.
# Ho: residuals have homoscedasticity

#If correction of robust standard errors is necessary
coeftest(model3, vcov = vcovHC(model3, type="HC0"))

#Multicollinearity Test (Only in cases of Multiple Regressions)
library(car)
vif(model3)
#VIF > 10.0 may indicate problems of multicollinearity

```

3 Submarines Regression

```
#Calling required packages
library("dplyr") #pacote para limpar as bases
library("ggplot2") #pacote para plotar gráficos
library("lmtest") #pacote para fazer testes de regressão multipla
library("sandwich") #pacote para calcular erros padrões robustos
library("readxl")
library("stargazer")

#csv file location for import
setwd("C:\\Users\\pulch\\Desktop\\Bases Reg\\Reg Sub")
getwd()

#Obtaining basic statistical data from the imported file
base <- read.csv2(file = "Data Reg Sub.csv", header = TRUE, dec = ",", sep = ";")
summary(base)

#Correlation analysis
cor(base)

#Running regression for 1st model
model1 <- lm(base$CST ~ base$LEN + base$BEAM + base$DIS.SUR, data= base)
summary(model1)
confint(model1)

#running regression for 2nd model
model2 <- lm(base$CST ~ base$LEN + base$BEAM, data= base)
summary(model2)
confint(model2)

#running regression for 3rd model
model3 <- lm(base$CST ~ base$LEN + base$DIS.SUR, data= base)
summary(model3)
confint(model3)

#running regression for 4th model
model4 <- lm(base$CST ~ base$DIS.SUR + base$BEAM, data= base)
summary(model4)
confint(model4)

#running regression for 5th model
model5 <- lm(base$CST ~ base$LEN, data= base)
summary(model5)
confint(model5)

#running regression for 6th model
```

```

model6<-lm(base$CST ~ base$BEAM, data= base)
summary(model6)
confint(model6)

#running regression for 7th model
model7<-lm(base$CST ~ base$DIS.SUR, data= base)
summary(model7)
confint(model7)

#generating summary report
stargazer(model1, model2, model3, model4, model5, model6, model7, out = "Sub Reg Result.html")

#choose model
#model 7

#regression graphic
#Creating data
df = data.frame(Displacement = base$DIS, Cost = base$CST)

grp <- ggplot(data = df,
             mapping = aes(x = Displacement, y = Cost)) +
  geom_point()+
  geom_smooth(method="lm", se=0) +
  labs(title="Regression Charts",
       subtitle="Chart 2 - Submarine's Regression") +
  ylab ("Cost ($M)") +
  xlab ("Displacement (t)") +
  theme_classic() +
  annotate("text",x=2500,y=1400,label = "p-value = 0.0002855", size = 3) +
  annotate("text",x=2500,y=1500,label = "R-squared = 0.5685", size = 3) +
  annotate("text",x=2500,y=1600,label = "Cost = 0.4059 * Displacement - 195.239")
grp

grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray") ,
        plot.subtitle = element_text(size = 10, color = "blue"))
grp

#residual normality test
#Shapiro-Wilk normality test
residuos<- model7$residuals
shapiro.test(residuos)
#p-value < 0.05 there is statistical evidence that leads to the rejection of H0
#H0: residuals follow normal distribution.
#However, by the central limit theorem, if n >= 30, one can relax with the assumption of normality of the residuals.

#heteroscedasticity test

```

```
#studentized Breusch-Pagan test
bptest(model7)
#p-value < 0,05 there is statistical evidence that leads to the rejection of H0.
# Ho: residuals have homoscedasticity

#If correction of robust standard errors is necessary
coefest(model7, vcov = vcovHC(model7, type="HC0"))
```

Appendix B: Monte Carlo Simulation Programming Script

1 Frigates Monte Carlo Simulation

```
#Installing and calling required packages
install.packages("extraDistr")
library("extraDistr")

#Creating vector ind_var to receive the triangular distribution of the independent variable
set.seed(999)
ind_var <- rtriang(10000, 4720, 5120, 4900)

#Calculating the mean
ind_var_mean <- mean(ind_var)
ind_var_mean

#Creating the lognormal distribution of the dependent variable
#Calculating the mean and standard deviation of the presumed normal distribution
x_mean <- 0.165 * ind_var_mean - 176.046
x_mean

x_sd <- 336.3
x_sd

#Calculating the mean and standard deviation of the lognormal distribution
y_mean <- 0.5 * log(x_mean ^ 4 / (x_mean ^ 2 + x_sd ^ 2))
y_mean

y_sd <- sqrt(log((x_mean ^ 2 + x_sd ^ 2) / (x_mean ^ 2)))
y_sd

#Creating the dep_var vector to receive the lognormal distribution
set.seed(999)
dep_var <- rlnorm(10000, meanlog = y_mean, sdlog = y_sd)

#Calculating quartiles of the lognormal distribution of the dependent variable
q80 <- qlnorm(0.8, meanlog = y_mean, sdlog = y_sd)
q80

#Creating Cost Distribution Chart
library(ggplot2)
grp <- qplot(dep_var, geom="histogram", fill=l("lightskyblue2")) +
  labs(title="Monte Carlo Simulation",
        subtitle="Frigates Cost Distribution Chart - 80%"
  ) +
  xlab("Cost (M$)") +
  ylab("Frequency") +
  annotate("Text", x=q80, y=1750,
```



```

      label = paste("80% = ",round(q80, digits=2)),
      col="black",
      size=2.5)
grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray") ,
        plot.subtitle = element_text(size = 10, color = "blue"))
grp

```

2 Patrols Monte Carlo Simulation

#Installing and calling required packages

```
install.packages("extraDistr")
```

```
install.packages("MASS")
```

```
library("extraDistr")
```

```
library("MASS")
```

#Creating vector ind_var to receive the triangular distribution of the independent variable

```
set.seed(999)
```

```
ind_var <- rtriang(10000, 1250, 1780, 1500)
```

#Calculating the mean

```
ind_var_mean <- mean(ind_var)
```

```
ind_var_mean
```

```
ind_var_sd <- sd(ind_var)
```

```
ind_var_sd
```

#Creating vector ind_var1 to receive the triangular distribution of the independent variable

```
set.seed(999)
```

```
ind_var1 <- rtriang(10000, 68, 74, 70)
```

#Calculating the mean

```
ind_var_mean1 <- mean(ind_var1)
```

```
ind_var_mean1
```

```
ind_var_sd1 <- sd(ind_var1)
```

```
ind_var_sd1
```

#Informing Correlation

```
cor_01 <- 0.8136
```

#Building the covariance matrix and generating the correlated normal random variable

```
cov_matrix <- matrix(c(ind_var_sd^2, cor_01*ind_var_sd*ind_var_sd1, cor_01*ind_var_sd*ind_var_sd1, ind_var_sd1^2),
  nrow=2)
```

```
set.seed(1)
```

```
mat_cor <- mvrnorm(20000, c(ind_var_mean, ind_var_mean1), cov_matrix)
```

```

ind_var_nmean <- mean(mat_cor[,1])
ind_var_nmean

ind_var_nmean1 <- mean(mat_cor[,2])
ind_var_nmean1

cor(mat_cor[,1], mat_cor[,2])

#Creating the lognormal distribution of the dependent variable
#Calculating the mean and standard deviation of the presumed normal distribution
x_mean <- 0.0256791 * ind_var_nmean + 0.887922 * ind_var_nmean1 -26.3216
x_mean

x_sd <- 41.22203
x_sd

#Calculating the mean and standard deviation of the lognormal distribution
y_mean <- 0.5 * log(x_mean ^ 4/(x_mean ^ 2 + x_sd ^ 2))
y_mean

y_sd <- sqrt(log((x_mean ^ 2 + x_sd ^ 2)/(x_mean ^ 2)))
y_sd

#Creating the dep_var vector to receive the lognormal distribution
set.seed(999)
dep_var <- rlnorm(10000, meanlog = y_mean, sdlog = y_sd)

#Calculating quartiles of the lognormal distribution of the dependent variable
q80 <- qlnorm(0.8, meanlog = y_mean, sdlog = y_sd)
q80

#Creating Cost Distribution Chart
library(ggplot2)
grp <- qplot(dep_var, geom="histogram", fill=l("mediumpurple2")) +
  labs(title="Monte Carlo Simulation",
        subtitle="Patrols Cost Distribution Chart - 80%"
  ) +
  xlab ("Cost (M$)") +
  ylab ("Frequency") +
  annotate("Text", x=q80, y=1750,
         label = paste("80% = ",round(q80, digits=2)),
         col="black",
         size=2.5)
grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray") ,
        plot.subtitle = element_text(size = 10, color = "blue"))

```

```
grp
```

3 Submarines Monte Carlo Simulation

```
#Installing required packages
install.packages("extraDistr")
library("extraDistr")

#Creating vector ind_var to receive the triangular distribution of the independent variable
set.seed(999)
ind_var <- rtriang(10000, 1960, 2350, 2150)

#Calculating the mean
ind_var_mean <- mean(ind_var)
ind_var_mean

#Creating the lognormal distribution of the dependent variable
#Calculating the mean and standard deviation of the presumed normal distribution
x_mean <- 0.4059 * ind_var_mean - 195.240
x_mean

x_sd <- 281.3
x_sd

#Calculating the mean and standard deviation of the lognormal distribution
y_mean <- 0.5 * log(x_mean ^ 4 / (x_mean ^ 2 + x_sd ^ 2))
y_mean

y_sd <- sqrt(log((x_mean ^ 2 + x_sd ^ 2) / (x_mean ^ 2)))
y_sd

#Creating the dep_var vector to receive the lognormal distribution
set.seed(999)
dep_var <- rlnorm(10000, meanlog = y_mean, sdlog = y_sd)

#Calculating quartiles of the lognormal distribution of the dependent variable
q80 <- qlnorm(0.8, meanlog = y_mean, sdlog = y_sd)
q80

#Creating Cost Distribution Chart
library(ggplot2)
grp <- qplot(dep_var, geom="histogram", fill=l("royalblue2")) +
  labs(title="Monte Carlo Simulation",
        subtitle="Submarines Cost Distribution Chart - 80%"
  ) +
  xlab ("Cost (M$)") +
```

```
ylab ("Frequency") +  
  annotate("Text", x=q80, y=1500,  
    label = paste("80% = ",round(q80, digits=2)),  
    col="black",  
    size=2.5)  
grp <- grp +  
  theme(plot.title = element_text(size = 15, color = "gray") ,  
    plot.subtitle = element_text(size = 10, color = "blue"))  
grp
```

Appendix C: Non-Linear Programming Script

1 Minimum Budget

```
#Setting costs of first units
cst_i_QF <- 852.1504
cst_i_QP <- 101.4689
cst_i_QS <- 876.1922

#Setting learning curve parameters
LC_QF <- 0.80
LC_QP <- 0.88
LC_QS <- 0.85

b_QF <- log(LC_QF)/log(2)
b_QP <- log(LC_QP)/log(2)
b_QS <- log(LC_QS)/log(2)

#creating a matrix to receive all possible quantities regarding restrictions
QF <- c(1:100)
QP <- c(1:100)
QS <- c(1:100)

nlinhas <- 0

mat_suc <- matrix(,3)
list_quant <- c()

for (i in 1:100) {
  for (j in 1:100) {
    for (k in 1:100) {
      if ((QF[i] + QS[k]) >= 16) {
        if ((QP[j] - 4 * QS[k]) >= 0){
          if ((QF[i] - 2 * QS[k]) >= 0) {
            if ((-2 * QF[i] + QP[j]) >= 0) {
              nlinhas = nlinhas + 1
              list_quant <- c(QF[i],QP[j],QS[k])
              mat_suc<-rbind(mat_suc,list_quant)
            }
          }
        }
      }
    }
  }
}

#Calculating cost
#Setting the maximum number of quantity
```

```

num_max <- 100

#creating a vector for unit cost
cst_QF <- c()
cst_QP <- c()
cst_QS <- c()

for (i in 1:num_max) {
  cst_QF[i] <- cst_i_QF * (i ^ b_QF)
  cst_QP[i] <- cst_i_QP * (i ^ b_QP)
  cst_QS[i] <- cst_i_QS * (i ^ b_QS)
}
print(cst_QF)
print(cst_QP)
print(cst_QS)

#creating a vector for cumulative cost
cst_QF_cum <- c()
cst_QP_cum <- c()
cst_QS_cum <- c()

for (i in 1:num_max) {
  if(i > 1) {
    cst_QF_cum [i] = cst_QF_cum [i - 1] + cst_QF [i]
  } else {
    cst_QF_cum [i] = cst_QF [i]
  }
}
print(cst_QF_cum)

for (j in 1:num_max) {
  if(j > 1) {
    cst_QP_cum [j] = cst_QP_cum [j - 1] + cst_QP [j]
  } else {
    cst_QP_cum [j] = cst_QP [j]
  }
}
print(cst_QP_cum)

for (k in 1:num_max) {
  if(k > 1) {
    cst_QS_cum [k] = cst_QS_cum [k - 1] + cst_QS [k]
  } else {
    cst_QS_cum [k] = cst_QS [k]
  }
}

```

```

print(cst_QS_cum)

#creating a cumulative cost column in the matrix that received all possible quantities regarding restrictions (success
matrix)
mat_suc1 <- mat_suc
new_mat <- cbind(mat_suc1,1)
new_mat <- cbind(mat_suc1,cst_QF_cum[new_mat[,1]] + cst_QP_cum[new_mat[,2]] + cst_QS_cum[new_mat[,3]])

#Finding minimum cost of success matrix
min_cst <- min(new_mat[,4], na.rm=T)
min_cst
which(new_mat == min_cst, arr.ind=TRUE)

#Finding QF, QP and QS responsible for the minimum cost
new_mat[1050,1]
new_mat[1050,2]
new_mat[1050,3]

#Shadow Prices
shp_QF <- cst_QF_cum[new_mat[1050,1]+1] - cst_QF_cum[new_mat[1050,1]]
shp_QF

shp_QP <- cst_QP_cum[new_mat[1050,2]+1] - cst_QP_cum[new_mat[1050,2]]
shp_QP

shp_QS <- cst_QS_cum[new_mat[1050,3]+1] - cst_QS_cum[new_mat[1050,3]]
shp_QS

```

2 Fixed Budget

```

#Setting costs of first units
cst_i_QF <- 852.1504
cst_i_QP <- 101.4689
cst_i_QS <- 876.1922

#Setting the available budget
avail_bud <- 32030

#Setting learning curve parameters
LC_QF <- 0.80
LC_QP <- 0.88
LC_QS <- 0.85

b_QF <- log(LC_QF)/log(2)
b_QP <- log(LC_QP)/log(2)
b_QS <- log(LC_QS)/log(2)

```

```
#creating a matrix to receive all possible quantities regarding restrictions
```

```
QF <- c(1:100)
```

```
QP <- c(1:100)
```

```
QS <- c(1:100)
```

```
nlinhas <- 0
```

```
mat_suc <- matrix(,,3)
```

```
list_quant <- c()
```

```
for (i in 1:100) {
```

```
  for (j in 1:100) {
```

```
    for (k in 1:100) {
```

```
      if ((QF[i] + QS[k]) >= 16) {
```

```
        if ((QP[j] - 4 * QS[k]) >= 0){
```

```
          if ((QF[i] - 2 * QS[k]) >= 0) {
```

```
            if ((-2 * QF[i] + QP[j]) >= 0) {
```

```
              nlinhas = nlinhas + 1
```

```
              list_quant <- c(QF[i],QP[j],QS[k])
```

```
              mat_suc<-rbind(mat_suc,list_quant)
```

```
            }
```

```
          }
```

```
        }
```

```
      }
```

```
    }
```

```
  }
```

```
}
```

```
#Calculating cost
```

```
#Setting the maximum number of quantity
```

```
num_max <- 100
```

```
#creating a vector for unit cost
```

```
cst_QF <- c()
```

```
cst_QP <- c()
```

```
cst_QS <- c()
```

```
for (i in 1:num_max) {
```

```
  cst_QF[i] <- cst_i_QF * (i ^ b_QF)
```

```
  cst_QP[i] <- cst_i_QP * (i ^ b_QP)
```

```
  cst_QS[i] <- cst_i_QS * (i ^ b_QS)
```

```
}
```

```
print(cst_QF)
```

```
print(cst_QP)
```

```
print(cst_QS)
```



```

#creating a vector for cumulative cost
cst_QF_cum <- c()
cst_QP_cum <- c()
cst_QS_cum <- c()

for (i in 1:num_max) {
  if(i > 1) {
    cst_QF_cum [i] = cst_QF_cum [i - 1] + cst_QF [i]
  } else {
    cst_QF_cum [i] = cst_QF [i]
  }
}
print(cst_QF_cum)

for (j in 1:num_max) {
  if(j > 1) {
    cst_QP_cum [j] = cst_QP_cum [j - 1] + cst_QP [j]
  } else {
    cst_QP_cum [j] = cst_QP [j]
  }
}
print(cst_QP_cum)

for (k in 1:num_max) {
  if(k > 1) {
    cst_QS_cum [k] = cst_QS_cum [k - 1] + cst_QS [k]
  } else {
    cst_QS_cum [k] = cst_QS [k]
  }
}
print(cst_QS_cum)

#creating a cumulative cost column in the matrix that received all possible quantities regarding restrictions (success
matrix)
mat_suc1 <- mat_suc
new_mat <- cbind(mat_suc1,1)
new_mat <- cbind(mat_suc1,cst_QF_cum[new_mat[,1]] + cst_QP_cum[new_mat[,2]] + cst_QS_cum[new_mat[,3]])

#creating a remaining budget column in the matrix that received all possible quantities regarding restrictions (success
matrix)
mat_suc2 <- new_mat
new_mat1 <- cbind(mat_suc2,1)
new_mat1 <- cbind(mat_suc2,avail_bud - cst_QF_cum[new_mat1[,1]] - cst_QP_cum[new_mat1[,2]] -
cst_QS_cum[new_mat1[,3]])

#Finding minimum cost of success matrix
min_cst <- min(new_mat1[,5][which(new_mat1[,5]>0)], na.rm=T)

```

```

min_cst

#Finding line of the minimum cost of success matrix
min_line <- which(new_mat1 == min_cst, arr.ind=TRUE)
lin <- min_line[1,1]
lin

#Finding QF, QP and QS responsible for the minimum cost
new_mat1[[lin,1]]
new_mat1[[lin,2]]
new_mat1[[lin,3]]

#Shadow Prices
shp_QF <- cst_QF_cum[new_mat1[[lin,1]]+1] - cst_QF_cum[new_mat1[[lin,1]]]
shp_QF

shp_QP <- cst_QP_cum[new_mat1[[lin,2]]+1] - cst_QP_cum[new_mat1[[lin,2]]]
shp_QP

shp_QS <- cst_QS_cum[new_mat1[[lin,3]]+1] - cst_QS_cum[new_mat1[[lin,3]]]
shp_QS

#CHARTS about shadow prices to the decision-makers
library(ggplot2)

#CHART 0 - Total Price, Shadow Price and Remaining Budget
# create a dataset
df <- data.frame(
  ships=c("Frigates","Patrols","Submarines"),
  value=c(cst_QF_cum[new_mat1[[lin,1]]], cst_QP_cum[new_mat1[[lin,2]]], cst_QS_cum[new_mat1[[lin,3]]])
)

#Creating chart 0
grp <- ggplot(data = df, aes(x=ships, y=value)) +
  geom_bar(stat="identity", fill="skyblue",
    colour="lightskyblue2") +
  labs(title="Decision-Maker Charts",
    subtitle="Chart 0 - Prices of Optimal Solution"
  ) +
  xlab ("Ships") +
  ylab ("Price ($M)") +
  annotate("Text", x="Frigates", y=16500,
    label = round(cst_QF_cum[new_mat1[[lin,1]]], digits=2),
    col="black",
    size=3) +
  annotate("Text", x="Patrols", y=5700,
    label = round(cst_QP_cum[new_mat1[[lin,2]]], digits=2),

```

```

      col="black",
      size=3) +
  annotate("Text", x="Submarines", y=12000,
    label = round(cst_QS_cum[new_mat1[[lin,3]], digits=2),
    col="black",
    size=3)

grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray") ,
    plot.subtitle = element_text(size = 10, color = "blue"))
grp

#CHART 1 - Total Price, Shadow Price and Remaining Budget
# create a dataset
data <- data.frame(
  ships=c("Frigates","Frigates","Patrols","Patrols","Submarines","Submarines") ,
  price=c("1 Shadow Price","2 Price","1 Shadow Price","2 Price","1 Shadow Price","2 Price") ,

value=c(shp_QF,cst_QF_cum[new_mat1[[lin,1]],shp_QP,cst_QP_cum[new_mat1[[lin,2]],shp_QS,cst_QS_cum[new_mat1[[lin,
3]])
)

#Creating chart 1
grp <- ggplot(data, aes(fill=price, y=value, x=ships)) +
  geom_bar(position="stack", stat="identity") +
  labs(title="Decision-Maker Charts",
    subtitle="Chart 1 - Price Analysis",
    fill = "Type of Price"
  ) +
  xlab ("Ships") +
  ylab ("Price ($M)")

grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray") ,
    plot.subtitle = element_text(size = 10, color = "blue"))
grp

#CHART 2 - Shadow Price and Remaining Budget
#Creating Data
ships=c("Frigates","Patrols","Submarines")
cost=c(shp_QF,shp_QP,shp_QS)
remaining=c(min_cst,min_cst,min_cst)
#Creating chart 2
grp <- ggplot()
grp <- grp + geom_bar(aes(x=ships, y=cost),stat="identity", fill="skyblue",
  colour="lightskyblue2")
grp <- grp + geom_line(aes(x=ships, y=remaining, group = 1),stat="identity") +

```

```

labs(title="Decision-Maker Charts",
      subtitle="Chart 2 - Shadow Price x Remaining Budget"
    ) +
xlab ("Ships") +
ylab ("Price ($M)") +
annotate("Text", x="Patrols", y=60,
         label = "Remaining Budget",
         col="black",
         size=5)
grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray"),
        plot.subtitle = element_text(size = 10, color = "blue"))
grp

#CHART 3 - Remaining Budget as a % of Shadow Price
#Creating data
data <- data.frame(
  ships=c("Frigates","Patrols","Submarines"),
  percentage=c(min_cst/shp_QF,min_cst/shp_QP,min_cst/shp_QS)
)
#Creating chart 3
grp <- ggplot(data = data,
              mapping = aes(x = ships, y = percentage)) +
  geom_bar(stat = "identity",
          color = "lightskyblue2",
          fill = "skyblue") +
  labs(title="Decision-Maker Charts",
        subtitle="Chart 3 - Remaining Budget / Shadow Price"
    ) +
xlab ("Ships") +
ylab ("Percentage of Shadow price") +
annotate("Text", x="Frigates", y=0.05,
         label = round(min_cst/shp_QF, digits=2),
         col="black",
         size=3) +
annotate("Text", x="Patrols", y=0.25,
         label = round(min_cst/shp_QP, digits=2),
         col="black",
         size=3) +
annotate("Text", x="Submarines", y=0.03,
         label = round(min_cst/shp_QS, digits=2),
         col="black",
         size=3)
grp <- grp +
  theme(plot.title = element_text(size = 15, color = "gray"),
        plot.subtitle = element_text(size = 10, color = "blue"))
grp

```

Appendix D: Reference Tables for Data Array

1 – Frigates Data

| Frigates | CST (\$M) | DIS (t) | RNG (NM) | LEN (m) | Source |
|------------------------------|-----------|---------|----------|---------|---|
| Anzac-class frigate | 707.18 | 3,600 | 6,000 | 118.00 | https://www.seaforces.org/marint/Australian-Navy/Frigate/Anzac-class.htm |
| Naresuan-class frigate* | 123.89 | 2,985 | 4,000 | 120.50 | https://www.naval-technology.com/projects/naresuan-class-guided-missile-frigates/ |
| Type 053H2 Jianghu-III class | 138.39 | 1,960 | 4,000 | 103.20 | https://www.globalsecurity.org/military/world/china/jianghu-specs.htm |
| Type 054A Jiangkai-II class | 246.00 | 4,053 | 8,025 | 134.10 | https://weaponsystems.net/system/419-Type-054A+class |
| Absalon class | 277.71 | 6,600 | 9,000 | 137.00 | https://www.naval-technology.com/projects/absalon/ |
| Iver Huitfeldt class | 393.16 | 6,645 | 9,300 | 138.70 | https://www.seaforces.org/marint/Danish-Navy/Frigate/Iver-Huitfeldt-class.htm |
| Aquitaine class | 923.38 | 6,000 | 6,000 | 142.00 | https://www.seaforces.org/marint/French-Navy/Destroyer-Frigate/Aquitaine-FREMM-class.htm |
| La Fayette class | 508.54 | 3,800 | 4,000 | 125.00 | https://www.seaforces.org/marint/French-Navy/Frigate-Corvette/La-Fayette-class.htm |
| Brandenburg class | 613.66 | 4,490 | 4,000 | 138.85 | https://www.naval-technology.com/projects/brand/ |
| F125 Baden-Württemberg class | 1,212.09 | 7,200 | 4,000 | 149.52 | https://www.seaforces.org/ |
| Sachsen class | 1,329.45 | 5,800 | 4,000 | 143.00 | https://www.naval-technology.com/ |
| Valour class | 443.08 | 3,700 | 8,000 | 121.00 | https://www.defencewebsite.co.za/resources/fact-files/fact-file-valour-class-small-guided-missile-frigates/?catid=79%3Afact-files&itemid=159 |
| Godavari class | 172.39 | 3,850 | 4,500 | 126.40 | https://naval-encyclopedia.com/cold-war/indian-navy# |
| Project 17A class | 1,029.80 | 6,670 | 1,000 | 149.00 | https://www.naval-technology.com/projects/nilgiri-class-project-17a-frigates/ |
| Shivalik class | 694.64 | 6,200 | 5,000 | 142.50 | https://www.naval-technology.com/projects/shivalik-class-friga/ |
| Carlo Bergamini class | 686.52 | 6,700 | 6,000 | 144.60 | https://www.seaforces.org/marint/Italian-Navy/Frigate/Bergamini-FREMM-class.htm |
| Maestrale class | 220.34 | 3,040 | 6,000 | 122.70 | https://www.seaforces.org/marint/Italian-Navy/Frigate/Maestrale-class.htm |
| De Zeven Provinciën class | 827.37 | 6,050 | 4,000 | 144.24 | https://www.naval-technology.com/projects/dezeven/ |
| Karel Doorman class | 324.85 | 3,320 | 4,500 | 122.00 | https://www.naval-technology.com/projects/karel/ |
| Fridtjof Nansen class* | 930.14 | 5,290 | 4,500 | 134.00 | https://www.seaforces.org/marint/Norwegian-Navy/Frigate/Fridtjof-Nansen-class.htm |
| Zulfiquar class* | 983.73 | 3,144 | 4,000 | 123.20 | https://quwa.org/2020/06/06/pakistan-navy-ships-zulfiquar-class-f-22p-frigate/ |
| Admiral Gorshkov class | 271.05 | 5,400 | 4,800 | 135.00 | https://www.naval-technology.com/projects/admiral-gorshkov/ |
| Admiral Grigorovich class | 626.66 | 4,000 | 4,850 | 124.80 | https://armedforces.eu/navy/frigate/Admiral_Grigorovich-class |
| Gepard class | 386.85 | 1,930 | 4,000 | 102.14 | https://www.naval-technology.com/projects/gepard-class/ |
| Talwar class | 674.27 | 4,035 | 4,850 | 124.80 | https://www.naval-technology.com/projects/talwarclassfrigate/ |
| Daegu class | 289.99 | 3,592 | 4,500 | 122.00 | https://www.seaforces.org/marint/Republic-Korea-Navy/Frigate/Daegu-class-FFG.htm |
| Incheon class | 274.31 | 3,251 | 4,500 | 114.00 | https://www.seaforces.org/marint/Republic-Korea-Navy/Frigate/Incheon-class-FFG.htm |
| Álvaro de Bazán class F101/4 | 709.43 | 5,800 | 4,500 | 146.70 | https://www.seaforces.org/marint/Spanish-Navy/Frigate/Alvaro-de-Bazan-class.htm |
| Álvaro de Bazán class F105 | 1,300.62 | 6,391 | 4,500 | 146.70 | https://www.naval-technology.com/projects/f100/ |
| Type 22 class Batch 3 | 188.88 | 5,300 | 7,500 | 148.10 | https://www.seaforces.org/marint/Royal-Navy/Frigate/Broadsword-Type-22-class.htm |
| Type 23 class | 439.39 | 4,900 | 7,500 | 133.00 | https://www.seaforces.org/marint/Royal-Navy/Frigate/Duke-Type-23-class.htm |
| Freedom class SSC | 395.76 | 3,500 | 3,500 | 115.00 | http://www.navyrecognition.com/index.php?option=com_content&view=article&id=72#specifications |
| Legend class cutter | 670.00 | 4,600 | 12,000 | 127.00 | https://www.dcms.uscg.mil/Our-Organization/Assistant-Commandant-for-Acquisitions-CG-9/Programs/Surface-Programs/National-Security-Cutter/ |
| Oliver Hazard Perry class | 164.50 | 4,200 | 4,500 | 138.00 | https://www.naval-technology.com/projects/oliver-hazard/ |
| Horizon class | 1,940.31 | 7,050 | 6,100 | 152.87 | https://www.seaforces.org/marint/French-Navy/Destroyer-Frigate/Forbin-HORIZON-class.htm |

2 – Patrols Data

| Patrols | CST (\$M) | DIS (t) | RNG (NM) | LEN (m) | Source |
|---|-----------|---------|----------|---------|---|
| Port of Spain class OPV | 83.40 | 2,200 | 5,500 | 90.50 | https://www.naval-technology.com/projects/amazonas-offshore-patrol-vessels/ |
| Armidale class | 30.80 | 300 | 3,000 | 56.80 | https://www.navy.gov.au/hmas-armidale-ii |
| ASMAR PZM class OPV | 70.80 | 1,850 | 8,600 | 80.60 | https://www.asmar.cl/download/opv.pdf |
| Buque de Accion Maritima class OPV | 190.70 | 2,860 | 8,700 | 93.90 | https://www.naval-technology.com/projects/maritimeactionship/ |
| Cape class | 32.90 | 300 | 4,000 | 57.80 | https://www.naval-technology.com/projects/cape-class-patrol-boats/ |
| Carnicobar class patrol boat - Fleet I | 8.20 | 293 | 2,000 | 48.90 | https://www.bharat-rakshak.com/NAVY/Ships/Active/201-Car-Nicobar-Class.html |
| Carnicobar class patrol boat - Fleet II | 8.20 | 315 | 2,000 | 48.90 | https://www.bharat-rakshak.com/NAVY/Ships/Active/201-Car-Nicobar-Class.html |
| Classe Yoon Youngha | 44.90 | 570 | 1,998 | 63.00 | https://www.seaforces.org/marint/Republic-Korea-Navy/Patrol-Vessel/Yoon-Youngha-class.htm |
| Clyde class OPV | 74.30 | 2,000 | 5,500 | 81.50 | https://www.naval-technology.com/projects/90m-offshore-patrol-vessel-opv/ |
| Comandanti-class | 112.70 | 1,520 | 3,500 | 88.60 | https://www.naval-technology.com/projects/commandante/ |
| Cyclone class patrol boat | 31.70 | 336 | 2,500 | 55.00 | https://warriorlodge.com/pages/cyclone-class-patrol-ship |
| Flyvefisken class | 31.60 | 450 | 3,860 | 54.00 | https://www.naval-technology.com/projects/fly/ |
| Harry DeWolf class OPV* | 341.70 | 5,800 | 6,800 | 97.00 | https://www.naval-technology.com/projects/harry-dewolf-class-arcticoffshore-patrol-ships-aops/ |
| Hero class | 20.90 | 257 | 2,000 | 42.80 | https://www.canada.ca/en/news/archive/2014/06/acceptance-cgcs-m-charles-m-b-new-hero-class-mid-shore-patrol-vessel-canadian-coast-guard-fleet.html |
| Holland class OPV | 213.80 | 3,750 | 5,000 | 108.40 | https://www.seaforces.org/marint/Netherlands-Navy/Offshore-Patrol-Vessel/Holland-class.htm |
| Holland-class offshore patrol vessel | 170.80 | 3,750 | 5,000 | 108.40 | https://www.seaforces.org/marint/Netherlands-Navy/Offshore-Patrol-Vessel/Holland-class.htm |
| Knud Rasmussen class OPV | 105.00 | 2,050 | 3,000 | 71.80 | https://www.seaforces.org/marint/Danish-Navy/Offshore-Patrol-Vessel/Knud-Rasmussen-class.htm |
| Meteoro-class offshore patrol vessel | 114.90 | 2,860 | 3,500 | 93.90 | https://www.buquesdeguerra.com/en/ships-spanish-navy/patrol-vessels/meteoro-class.html |
| Orca class | 11.00 | 210 | 660 | 33.00 | https://www.naval-technology.com/projects/orca-class/ |
| Pacific class | 11.30 | 162 | 2,500 | 31.50 | Blewett, Daniel K. RQ 36, no. 3 (1997): 467–68. http://www.jstor.org/stable/20863194 . |
| Pattani class OPV | 68.80 | 1,460 | 3,500 | 95.50 | https://shipshub.com/classes/142-2.html |
| Protector class OPV | 75.90 | 1,900 | 6,000 | 85.00 | https://www.naval-technology.com/projects/protectorclassoffsho/ |
| Rotoiti class IPV | 23.10 | 340 | 3,000 | 55.00 | https://www.seaforces.org/marint/New-Zealand-Navy/Protector-Rotoiti-class-IPV.htm |
| Rubin class patrol boat | 40.20 | 630 | 3,500 | 62.50 | https://www.homelandsecurity-technology.com/projects/rubin-class-patrol-boats-russia/ |
| Samuel Beckett class | 108.60 | 2,256 | 6,000 | 90.00 | https://www.naval-technology.com/projects/samuel-beckett-class-offshore-patrol-vessels-opvs/ |
| Sankalp-class | 137.40 | 2,020 | 6,500 | 105.00 | https://www.janes.com/defence-news/news-detail/indian-coast-guard-takes-delivery-of-another-sankalp-class-opv |
| Sentinel class | 70.20 | 359 | 2,500 | 46.80 | https://www.navalnews.com/naval-news/2020/09/four-more-sentinel-class-fast-response-cutters-for-the-us-coast-guard/ |
| Sirio class OPV | 93.40 | 1,549 | 330 | 88.60 | https://www.fincantieri.com/en/products-and-services/naval-vessels/pattugliatori-daltura/sirio-class/ |
| Stan Patrol 4207 | 10.20 | 239 | 2,450 | 42.80 | https://www.naval-technology.com/uncategorised/newsdamen-to-construct-three-more-stan-patrol-4207-patrol-vessels-for-mexican-navy-4771424/ |
| Svalbard class OPV | 88.60 | 6,375 | | 103.70 | https://www.homelandsecurity-technology.com/projects/kv-svalbard-w-303-icebreaker-opv/ |
| Turva class OPV | 151.70 | 4,000 | | 95.90 | https://www.homelandsecurity-technology.com/projects/turva-offshore-patrol-vessel-finland/ |
| Viana Do Castelo class OPV | 70.80 | 1,600 | | 83.10 | https://www.naval-technology.com/projects/viana-do-castelo-class-offshore-patrol-vessels/ |
| Yoon Youngha-class patrol vessel | 50.20 | 570 | 1,998 | 63.00 | https://www.seaforces.org/marint/Republic-Korea-Navy/Patrol-Vessel/Yoon-Youngha-class.htm |

3 – Submarines Data

| Submarines | CST (\$M) | LEN (m) | BEAM | DIS SUR (t) | Source |
|---------------------------------|-----------|---------|-------|-------------|---|
| Collins class SSK | 1,299.14 | 77.42 | 7.80 | 3,100 | https://www.seaforces.org/marint/Australian-Navy/Submarine/Collins-class.htm |
| Type 035 Ming class SSK | 113.87 | 76.00 | 7.60 | 1,584 | https://www.naval-technology.com/news/myanmar-commissions-type-35b-submarine/ |
| Type 041 Yuan class SSK | 683.29 | 77.60 | 8.40 | 2,700 | https://news.usni.org/2015/08/31/essay-inside-the-design-of-chinas-yuan-class-submarine |
| Agosta 90B class SSK | 845.59 | 76.20 | 6.80 | 1,570 | https://militaryanalyzer.com/agosta-class-submarine/ |
| Scorpène CA-2000 class SSK | 590.24 | 61.70 | 6.20 | | https://www.naval-technology.com/projects/scorpene-class/ |
| Dolphin 2 class SSK | 753.48 | 68.60 | 6.80 | 2,050 | http://www.military-today.com/navy/dolphin_class.htm |
| Shishumar class (Type 209/1500) | 872.18 | 64.40 | 6.50 | 1,660 | https://www.nti.org/wp-content/uploads/2021/09/shishumar_type_209_1500.pdf |
| Heroine class (Type 209/1400) | 362.58 | 62.00 | 7.60 | 1,545 | https://web.archive.org/web/20080917205630/http://www.navy.mil.za/archive/0808/080819_Fleet_review/heroine_submarines.htm |
| Type 210 'Ula' class SSK | 147.50 | 59.00 | 5.40 | 1,040 | https://www.naval-technology.com/projects/ula-class-submarines/ |
| Type 212A class SSK | 655.82 | 56.00 | 6.80 | 1,524 | https://www.seaforces.org/marint/Italian-Navy/Submarine/Todaro-Type-212-class.htm |
| Type 214 class SSK | 399.21 | 65.00 | 6.30 | 1,690 | https://www.seaforces.org/marint/Hellenic-Navy/Submarine/Papanikolis-class.htm |
| Type 216 class SSK | 1,770.41 | 90.00 | 8.10 | 4,000 | http://www.navyrecognition.com/mobile/index.php/oceania/australia/submarines/264-type-216-u-216-conventional-submarine-ssk-aip-tkms-hdw-submarine-class-216-howaldtswerke-deutsche-werft-thyssenkrupp-marine-systems-royal-australian-navy-datasheet-pictures-i |
| Söryü class | 602.11 | 84.00 | 9.10 | 2,900 | https://www.naval-technology.com/projects/ssoryuclasssubmarin/ |
| Walrus class | 467.63 | 67.73 | 8.40 | 2,465 | https://www.seaforces.org/marint/Netherlands-Navy/Submarine/Walrus-class.htm |
| Kilo class | 347.27 | 73.80 | 9.90 | 2,350 | https://www.naval-technology.com/projects/kilo/ |
| Jang Bogo class | 413.45 | 61.00 | 6.30 | 1,180 | https://www.seaforces.org/marint/Republic-Korea-Navy/Submarine/Jang-Bogo-class.htm |
| S-80 class | 1,180.63 | 81.05 | 11.68 | 3,200 | https://www.navyrecognition.com/index.php/naval-news/naval-news-archive/2021/july/10488-spanish-isaac-peral-s-80-class-submarine-takes-next-step.html |
| Blekinge class | 524.22 | 63.00 | 6.40 | 1,800 | https://www.navalnews.com/naval-news/2021/08/saab-awarded-contract-to-add-capabilities-on-future-swedish-submarines/ |

References

- Balakrishnan, Nagraj, Barry Render, Ralph M. Stair, and Chuck Munson. 2017. *Managerial Decision Modeling*. Boston: Degruyter.
- Foundation, R. n.d. *What is R?* Accessed January 18, 2022. <https://www.r-project.org/about.html>.
- Ghanmi, Abderrahmane Sokri and Ahmed. 2017. "Cost RiskAnalysis and Learning Curve in the Military Shipbuilding Sector." *Defence Research and Development Canada* 6.
- Hair, Joseph F., Anderson, Rolph E., Black, William C. 2014. *Multivariate Data Analysis*. Tulsa: PPC Books.
- Mislick, Gregory K., and Daniel A Nussbaum. 2015. *Cost Estimation: Methods and Tools*. Hoboken, NJ: Wiley & Sons.
- Thomas, Duncan, and John Fitch. 2014. "Joint Agency Cost Schedule Risk and Uncertainty Handbook." Washington D.C.: U.S. Government.
- W.N. Venables, D.M. Smith, R Core Team. 2021. *An Introduction to R*. 08 10.