CARLETON UNIVERSITY

SCHOOL OF
MATHEMATICS AND STATISTICS

HONOURS PROJECT

TITLE: Using a Double Dummy Solver and Multiple Linear Regression to Determine a Ranking System for Players of the Card Game of Bridge

AUTHOR: Benjamin Marriner

SUPERVISOR: Jason Gao

DATE: January 10th, 2024

Using a Double Dummy Solver and Multiple Linear Regression to Determine a Ranking System for Players of the Card Game of Bridge

Acknowledgements

Firstly, I would like to thank Jason Gao for supervising the project, introducing me to bridge, and proposing the idea of using multiple linear regression to determine a ranking system for players.

I would also like to thank Bo Haglund for creating the Double Dummy Solver program and making it available for public use along with extensive documentation.

Finally, I would like to thank Richard Van Haastrecht, for providing the data file and answering my questions regarding it.

Abstract

Bridge is a card game of skill. The World Bridge Federation maintains lists that rank bridge players by their skill. The ranking system involves awarding points based on successful outcomes in tournaments. In this research, an attempt is made to devise a new ranking system based on every aspect of a player's game. A file containing bridge boards is first cleaned, and the valid boards are extracted for use in the analysis. Players listed in the file are matched to names that appear on the world ranking lists. A double dummy solver, which is a computer program that can determine the optimal play in any scenario, is then used to analyse the boards. Players are assigned a set of unweighted variables that describe the number of times that their play differed from that of the double dummy solver. Players are also assigned a set of weighted variables that take into account the differences in score earned by the player and by the double dummy solver in each scenario. It is expected that these variables will all be negatively correlated with the number from the world rankings that determines a player's ability. If a linear regression model can be fit, the line of best fit can be used as a function of the variables collected to determine the new ranking system. The results of the double dummy analysis are first reduced to include only players who participated in a sufficient number of boards. Then, the variance inflation factor is computed for each independent variable to ensure that there is no multicollinearity. The result of this is that all of the variance inflation factor values are small and all variables are considered in the regression model selection procedure. The procedure resulted in the chosen model depending only on one variable, the percentage of cards played as a defender that were suboptimal. The variable had a coefficient of -1028.60322 and the intercept of the model was 35.43033. The model had an R^2 value of only 0.0693 and was found to violate the assumptions of linear regression. Thus, it was concluded that the variables collected were not correlated with the world rankings, and so a new ranking system was not able to be devised.

Table of contents

1. Introduction

The desire to rank players of the card game of bridge follows naturally from the fact that bridge, when played in its duplicate variety, is entirely a game of skill. The governing body for bridge, the World Bridge Federation (WBF), currently defines five ranking categories and two ranking systems. The five categories are open, women's, mixed, youth, and senior. A WBF sanctioned tournament will be contested by players who belong to one of these five categories. Prior to a tournament, the WBF will specify the number of masterpoints and the number of placing points that will be awarded to participants based on their performance. More prestigious tournaments that attract a higher standard of player will award more masterpoints and more placing points to the best-performing players. These points are specific to each category; that is, open masterpoints differ from mixed masterpoints, and so on. The two types of points differ in the manner that masterpoints represent the current ability of a player, whereas placing points represent the career of a player. Masterpoints are awarded more liberally, and decay at a rate of fifteen percent per year. Placing points are awarded more sporadically, but do not decay. This ranking framework thus admits ten distinct rankings, determining the current best player and the all-time best player in each of the five categories.

This research aims to provide an alternative ranking system for bridge players based on the smallest units of measure of one's play. A computer program will be devised that will take as input, records of a player's bridge boards, and compare their play in each board to that of a double dummy solver (DDS). A DDS is itself a computer program that operates on full information of each player's hand. It uses this information to determine the optimal play in any scenario. The differences between the outcomes of a given player's decisions and those of the DDS will then be expressed as a set of variables, judging the player on all aspects of the game. This collection of values will serve as a measure of merit for the player. To determine which subset of these variables is most accurate at predicting a player's ability, and to what extent each variable has an effect, multiple linear regression modelling will be performed. The independent variables collected from the double dummy analysis for each player will be compared to their current world ranking. If the established model is useful, the resulting line of best fit can be used as a function to determine a single value representing a player's ability. Players may then be ranked by this value.

Readers unfamiliar with the rules of bridge should consult Appendix A: The Rules of Bridge. Readers should also be familiar with multiple linear regression before continuing. Concepts discussed include variance inflation factor, testing assumptions regarding the error terms, and the model selection techniques of stepwise regression, forward selection, and backward elimination. [3] provides comprehensive coverage of these topics. All external links in this paper, aside from the references, are links to a GitHub repository containing files relevant to this research.

2. Methodology

2.1. Data Cleaning

This research requires a large dataset of bridge boards that are contested by players who appear in the current world rankings. The largest dataset that was able to be obtained was one provided by Richard Van Haastrech, containing about 200 000 boards from many major tournaments between 1955 and 2016. The original file is here. The file itself is structured similarly to that of an SQL insert script, however there are issues that prevent it from being read into a database in its initial form. The file uses semicolons to separate the data entries for a board, but then also uses semicolons within the data entries. So, there is no token to split the strings on to extract the values of each variable. Semicolons are also used at the end of each line, sometimes more than one and sometimes none, as opposed to the comma that SQL requires. Each data entry is wrapped in single quotes, however single quotes are also used in names, sometimes escaped and sometimes not. The file also contains some other oddities that disallow it from conforming to the SQL standards. So, a series of manipulations were performed to clean the file and allow it to be read into an SQLite3 database. The steps that were taken follow, with character sequences bolded. The modified file is here.

1. Change the semicolon separators in line 1 to commas.

2. Remove the 7 semicolons after **VALUES** on the first line.

3. Remove line 150242.

4: Add a comma to the end of lines 166797, 186780 and 186802 (these are the updated line numbers following the previous steps).

5. Replace all runs of 7,6,5,4,3,2 and 1 semicolons preceded by a **')** with a **'),** , in that order.

6. Replace

##

###

###

###################### with **''** .

7. Remove all **''** .

8. Replace all **'** with **''** .

9. Replace **'';''** with **','** .

10. Replace **''),** with **'),** .

11. Replace ***;''** with ***,'** , where * is any number between 0 and 9 inclusive.

12. Remove all \\ .

13. Replace the **,** at the end of the last line with **;** .

Each board in the file is defined by a set of variables. Each of these variables are described in Table 1. The information for each board was recorded by hand by annotators who were physically present at the table. As such, some of the values recorded may be nonsensical or an obvious typo. These erroneous values are reflected in Table 1.

Table 1

Variables defining a board in the original file

Variable	Description	Possible Values	Notes
event	The name of the tournament the board was a part of.	Any string.	Often contains the year the tournament was played.
date	The date the board was played.	One of [“”,”?”,”#”] or In the form YYYY.MM.DD.	Occasionally uses one of [“??”,”?”,””] in place of MM or DD.
site	Where the board was played.	Any string.	Not relevant to this research.
board	The number of this board within this match.	Natural numbers.	Not relevant to this research.
stage	A description of the stage of the tournament.	Any string.	Not relevant to this research.
dealer	The player who dealt the cards.	One of [“N”, “E”, “S”, “W”, “”]	Typically the one who starts the bidding.
deal	The cards that each player has.	Empty, or in PBN format.	Example: N:AK75.54.987653.A Q.AT983.42.JT753 642.KQJ7.AQJ.962 JT983.62.KT.KQ84
vulnerable	Which partnerships are vulnerable.	One of [“All”,”None”,”EW”,”NS”,”NZ”,”All]”,””]	“NZ” is likely a typo for “NS”, and “All]” for “All”.
bidstart	The player who actually started the bidding for this board.	One of [“N”, “E”, “S”, “W”,“9”,”3”,”7”,”4”,“”]	Can conflict with dealer. Values such as “7” are nonsensical.
auction	A space-separated sequence of bids in order, starting with the bid made by bidstart.	Any string.	Often contains invalid bids.

Table 1 (Continued)

contract	The contract to be attempted by the declarer.	One of [“”,”Pass”,”pass”] or any valid contract.	Uses “X” for double, “XX” for redouble, “NT” for no trump. Example: “7NTXX”.
lead	The player who leads to the first trick.	One of [“N”, “E”, “S”, “W”, “”]	Nothing to note.
play	A space-separated sequence of the cards played.	Any string.	Fewer than 52 cards played indicates a claim occurred.
declarer	The declarer for this board.	One of [“N”, “E”, “S”, “W”,”^S”,“”]	“^S” is likely a typo for “S”, or the indication of an unusual situation.
result	The total number of tricks taken by the declarer.	Natural number between 0 and 13 inclusive. Also one with 14.	The value of 14 is obviously an error.
note	Additional information about the board.	Any string.	Not relevant to this research.
westname	The name of the player sitting in the west seat.	Any string.	The same player may have their name represented differently in two different boards.
eastname	The name of the player sitting in the east seat.	Any string.	Same note as Westname.
northname	The name of the player sitting in the north seat.	Any string.	Same note as Westname.
southname	The name of the player sitting in the south seat.	Any string.	Same note as Westname.
westpoints	The number of high card points that west has.	Natural numbers.	Not relevant to this research.
northpoints	The number of high card points that north has.	Natural numbers.	Not relevant to this research.
eastpoints	The number of high card points that east has.	Natural numbers.	Not relevant to this research.

Table 1 (Continued)

southpoints	The number of high card points that south has.	Natural numbers.	Not relevant to this research.
scoring	The scoring system that was used.	One of [“IMP”, “MP”, “Chicago”, “IMP;Cross”, “BAM”, “Imp”, “”, “#”]	Not relevant to this research.
home_team	The team designated as the home team for the match.	Any string.	Not relevant to this research.
visit_team	The team designated as the away team for the match.	Any string.	Not relevant to this research.
room	Whether the board was played in the open room or the closed room.	One of [“Open”, “Closed”, “”]	Not relevant to this research.
description	Additional information about the board.	Any string.	Not relevant to this research.

The SQL script used to create the table that will hold the boards is here. The database file itself is here.

The boards may now be selected from the database and cleaned. Particular variables are crucial to performing any analysis at all, and if a board is missing a value for such a variable, or the value of such a variable is invalid, the board will be discarded. Furthermore, a board will also be removed from consideration if particular sets of variables contradict each other. This is possible as some variables are deducible from the values of other variables. The cleaned boards will be supplied to the DDS, which will perform an analysis of the two main portions of the game: the auction and the play. It is possible for one of these variables to be valid while the other is not. In such cases, the value of the invalid variable will be set to an empty string. This will signify to the double dummy analysis program not to perform the analysis on that portion of the game for that particular board. The board is still included as the other aspect of play may still be analysed. Only some of the variables mentioned in Table 1 will be verified, specifically the ones that do not state they are irrelevant to this research. Invalidities or inconsistencies in these irrelevant variables are not grounds for removing the board from the dataset and thus are not investigated. The cleaned boards that will be sent to the double dummy analysis program consist of some of the variables from Table 1 with cleaned values, along with new variables that will aid in the double dummy analysis.

The ***deal*** variable is checked for validity by ensuring that the PBN format supplied describes a scenario in which every player has 13 cards, every card from the standard 52 card deck appears exactly once, and that every card is itself a valid card from the standard deck. The *deal* variable must not be empty or invalid, otherwise the board will be excluded.

To validate the ***auction*** variable, the bids listed in the auction are first cleaned. The set of all bids that were made in any board were obtained from the database, and are here. Some of these bids are valid, some are invalid but recognizable and normalizable, and some are unrecognisable. The unrecognised bids are simply ignored, and the set of ignored bids is here. The cleaned auction must now follow the rules stipulated by the laws of bridge, otherwise the *auction* variable will be set to an empty string. Firstly, three passes must appear exactly once at the end of the auction, with the exception of three passes to start the auction. Secondly, every contract bid made must be in an increasing fashion. Thirdly, any double bid must have been made by the side who did not bid the previous contract, and the previous contract must not have already been doubled. Finally, any redouble bid must be made by the side who did bid the most recent contract, and the contract must have already been doubled. If the *auction* variable was already empty, no action is taken.

Similarly to the *auction* variable, the ***play*** variable is first cleaned by cleaning the plays that it is made of. The set of all plays is here. Again, most plays are valid or at least recognizable. The set of ignored plays is here. Within each trick, the play sequences from the database are always listed in clockwise order starting with the original leader. If a claim was made in the middle of a trick, dashes are used in place of the cards that had not yet been played, to preserve the aforementioned order. Thus, the number of plays in the play sequence should always be divisible by four. A list of the players that played each card in the order that they did is computed and returned as a new variable, called *play_order*. Once the card order has been modified to reflect the actual order the cards were played, no dash should appear at the start of a trick and no dash should follow a card in a trick. Furthermore, dashes should appear in at most one trick. No player should ever play the same card twice, or play a card that they did not have to begin with. Finally, no player should renege. If any of the above assertions are false, the play sequence is invalid and the *play* variable will be set to an empty string. Otherwise, the *play* variable is set to the string of cards played in their true order, without dashes. If the *play* variable, the *lead* variable, or the *contract* variable are empty, the *play* variable will be set to an empty string.

The ***date*** and ***event*** variables are together used to determine the value of a new variable to be used in later analyses, the *year* variable. The *date* variable is searched for a number that appears to be a year. If a year candidate is not found in the *date* variable, the *event* variable is searched. If a year candidate is found, it must fall within the known date range of the database, which is between 1955 and 2016, inclusive. If no year is found, the *year* variable is set to 0.

The ***vulnerable*** variable is normalised by assuming a value of "All]" intended to specify "All", and a value of "NZ" was intended to be "NS". The *vulnerable* value is then used to create two new boolean variables, *ew_vulnerable* and *ns_vulnerable*. If the *vulnerable* variable is empty, it is assumed that neither side is vulnerable.

The ***result*** variable is used to find the value of a new variable, the *claim* variable. If the *result* variable contains a value less than zero or greater than 13, *claim* is set to -1. Otherwise, *claim* is set equal to the value of the *result* variable. If the *play* variable does not contain a full play sequence, a claim must have occurred. The *result* variable contains the total number of tricks won by the declarer's side at the conclusion of the board. If the *play* variable, in conjunction with the *declarer* and *contract* variables, describes a scenario in which the declarer has already taken more tricks than the number specified by *result*, this is a contradiction. Moreover, if the value of *result* is greater than the sum of the number of tricks won by the declarer in the play sequence, and the number of tricks remaining to be played, this is also a contradiction. In the event of one of these contradictions, the board is discarded.

The ***dealer*** and ***bidstart*** variables are used to compute the cleaned value of the *bidstart* variable. If both values are not empty and they conflict, *bidstart* is set to the current value of the *bidstart* variable. If one of the variables is empty, *bidstart* is set to the value of the non-empty variable. It is always the case in the data that if neither *dealer* nor *bidstart* is given, the *auction* is also not given. Since *bidstart* is only used by the double dummy analysis program to analyse the auction, it can be left empty when *auction* is empty.

If each of ***northname***, ***eastname***, ***westname*** and ***southname*** is empty, then the board is discarded. If not, each of these name variables must be set to a name that appears in the world rankings, or an empty placeholder name if that player does not appear on the world rankings. Ideally, the relevant name variable is set precisely to the player on the world rankings list who actually played the board. However, as the names in the database may differ from the name used in the world ranking, and in fact may even differ between database entries, it is not possible to directly compare them for equality. Instead, a fuzzy matching algorithm will be used, which may be inaccurate in some cases. The algorithm first computes, for every name in the database and every name in the world rankings, a collection of names and a collection of initials. The collection of names for a given player could contain the player's first name, last name, middle name, nicknames, etc. To determine which name from the world rankings will be matched to a given database name, a particular tuple of four values is computed for each world ranking name. The first and most significant value is the number of entries in the names collection of the database name that can be matched to the names collection of the world ranking name. The second value is the number of initials that can be matched. An initial can be matched to another initial, or to an unmatched name in the names collection of the world ranking name, should the entry start with the same letter as the initial. The third value of the tuple is the number of entries in the names collection of the world ranking name that were not matched. The fourth value is the number of initials of the world ranking name that were not matched. The third and fourth values are given negative weight so that a smaller number of unmatched entries results in a larger number. The tuples are then sorted lexicographically. The world ranking name associated with the maximum tuple is the name that is assigned to the relevant name variable, with a few exceptions. Firstly, the first value in the tuple must be at least one, otherwise no match is made. Secondly, if there are multiple maximum tuples, no additional tiebreakers are enforced and there will instead be no match. If there is no match, the relevant name variable is set to an empty placeholder name that will be removed from later analysis.

The cleaned ***declarer*** value is set to the value of the *declarer* variable if it is one of "N","E","S", or "W". The declarer of a board is also determinable using the *auction* and *bidstart* variables. If the declarer that is deducible from these variables contradicts the value specified in the *declarer* variable, the board is discarded. If the *auction* variable describes a board that was passed out, it asserts that the value of *declarer* does not contain any value, and sets the value of *declarer* to "P".

The ***lead*** variable must specify the player who sits to the left of the declarer. If the *lead* variable is non-empty and specifies a player who does not sit to the left of the declarer, the entire board is discarded on the basis of a contradiction. If the *lead* is empty, it is set to the appropriate value. The *lead* variable is set to "P" if it is empty and *declarer* is "P".

To determine the cleaned value of the ***contract*** variable, the contents of the *contract* variable are first normalised. No trump contracts will now be represented by "N" instead of the previous "NT". Redoubled contracts will be represented by "R" instead of "XX". All passed contracts become "P". If the *contract* variable contradicts the contract deducible from the *auction*, then the board is discarded. Furthermore, if *contract* is "P" and the *play* variable is non-empty, the board is discarded.

A summary of the variables used to define each clean board is provided in Table 2.

Table 2

Variables defining a clean board

Variable	Description
year	The year the board was played in, or 0.
northname	A name from the world rankings list.
eastname	A name from the world rankings list.
soutname	A name from the world rankings list.
westname	A name from the world rankings list.
ew_vulnerable	True if East and West are vulnerable, false otherwise.
ns_vulnerable	True if North and South are vulnerable, false otherwise.
deal	The hand of each player, in PBN format.
bidstart	The player who started the bidding. One of [“N”,”E”,”S”,”W”], or an empty string.
auction	A space-separated string of valid bids, or an empty string if invalid.
contract	In the form LSD: where L is one of [1 - 7], S is one of [“C”,”D”,”H”,”S”,”N”], and D is one of [“”,”X”,”R”]. Or, “P” or an empty string.
declarer	The declarer. One of [“N”,”E”,”S”,”W”,”P”].
lead	The lead. One of [“N”,”E”,”S”,”W”,”P”].
play	A string of valid cards in the order they were played.
play_order	A list of integer representations of the players, ordered in a way so that the player in the i^{th} position played the card in the i^{th} position of *play*.
claim	-1 if no claim was made, otherwise equal to the number of tricks the declarer won in total.

The code that performs these cleaning processes is here. All code is implemented in Python. The output of this program can then be used as the input to the double dummy analysis program.

2.2. Double Dummy Analysis

The double dummy analysis program will use a DDS developed by Bo Haglund. The program will make use of two particular functions of this DDS. The first takes as input, in addition to the other variables that define a board, the sequence of cards that were played. From the previous data cleaning output, this is stored in the *play* variable. The DDS then outputs a list of double dummy values, of which there is one more than the number of cards played in the provided sequence. There is one double dummy value for after each card is played, and the first double dummy value is for before any cards were played. The double dummy value is equal to the number of tricks that the declarer's side can win, assuming that the remainder of the cards to be played will be played perfectly by each side. If the declarer plays a card, and the double dummy value decreases, the declarer has made a mistake. The difference between the new dummy double value and the previous double dummy value is equal to the number of tricks that the declarer may have cost their side by playing the card they did and not the optimal card. If the defender plays a card and the double dummy value increases, the defender has made a mistake. It is not possible for the double dummy value to increase following the declarer's play, or to decrease following a defender's play. The second function of the DDS that will be utilised takes only the *deal* variable. It returns a table specifying, for each suit, the number of tricks that each player could win if they became the declarer, assuming all cards are played perfectly by each side.

To analyse the auction or play segments of a board, score computations will need to occur. A particular element of the scoring, the number of penalty points awarded for the fourth and subsequent undertricks in a non-vulnerable, doubled or redoubled contract, changed after 1987. Thus, the score computations are dependent on the year the board was played. If the *year* variable from the data cleaning output was 0, the scoring method from after 1987 is used.

If the *play* variable of the board is non-empty, the players involved in the board can be judged on their play. First, the DDS is supplied with the *play* variable to obtain the double dummy values. Then, for each card a player played, it can be determined whether or not they had made a mistake. As the challenge of playing the optimal card differs based on a player's position, the variables collected are split positionally. The unweighted mistake variables are *lead_mistakes*, *play_mistakes_as_declarer*, and *play_mistakes_as_defender*, which are incremented by 1 if the player makes a mistake from the lead, declarer, or defender positions respectively. Variables *leads*, *cards_played_as_declarer*, and *cards_played_as_defender* are also tracked. Thus one could find the percentage of time that a player makes a particular type of mistake, by dividing the relevant number of unweighted play mistakes by the number of cards played from that position. Weighted mistakes are also computed, which take into account the magnitude of the mistake. The magnitude of a mistake is computed as the difference in score between the amount of points the side stood to win before the mistake was made, and the amount of points they now stand to win. The score computations consider the relevant side's vulnerability, as stipulated by the laws of bridge. These weighted values are stored in the variables *weighted_lead_mistakes_per_lead*, *weighted_play_mistakes_as_declarer*, and *weighted_play_mistakes_as_defender*, respecting the position of the player when the mistake was made. Finally, the claims players make can be evaluated by comparing the value of the *claim* variable from the data cleaning procedure to the final double dummy value, assuming the play sequence was incomplete and thus a claim was made. The final double dummy value represents the number of tricks the declarer would have won had the rest of the cards been played out, assuming optimal play from each side. The *claim* variable is the total number of tricks the declarer has taken. If the *claim* variable is larger than the final double dummy value, the claim favoured the declarer. If it was smaller, the claim favoured the defenders. If it is the same, no mistake was made by either side. The total number of claims a player is involved in is stored in the *claims* variable, the number of times they made a mistake is stored in *claim_mistakes*, and the weighted cost of all of their claim mistakes is stored in *weighted_claim_mistakes*.

If the *auction* variable is non-empty, the players may be judged on their performance in the auction. The number of auctions a player is a part of is stored in *auctions_analyzed*. Auction performance is judged as a partnership, unlike play performance which is judged individually. The DDS is provided with the *deal* variable, and so it returns the table specifying how many tricks each player could win as declarer in each suit. The analysis is then split into three categories, depending on whether a given player was on the defending side of the eventual contract, the declaring side, or if the contract was passed out. In any of these cases, the analysis considers all contracts that a partnership had available to them. A contract is available to a partnership if the opponents did not make a bid which would make it impossible for the partnership to bid the contract.

Firstly, if the deal is passed out, all contract bids were available to both partnerships, with any player as declarer. The score of each possible contract is considered and compared to the score that the partnership would earn from the current contract, which is 0, as the contract was passed. The score of a prospective contract is computed using the double dummy value that specifies the number of tricks a given declarer could win in a given suit, under the assumption that both sides play the cards optimally. If there is an available contract that would have provided a positive score to a side had they bid it, a mistake has been made by that side. The variable *passed_higher_contract* is incremented, and the variable *weighted_passed_higher_contract* is increased by the highest score that could have been earned had the partnership moved into a higher contract.

If the partnership currently under scrutiny is the defenders in the current contract, there are three types of contracts that could have been available to them. The first is any contract higher than the current contract. The second is doubling the current contract, if it is not already doubled. The third is not doubling the current contract, had it been doubled. Any higher contract that the defenders could move into but cannot make will be assumed to be doubled by the opponents. It is possible that only one of the defenders could be the declarer in particular suits for any higher contract they could move into, and this is taken into account. The highest score possible from each of the applicable scenarios is computed, and compared to the current score. If the highest score from any of the scenarios is better than the current score, a mistake has been made. The scenario that admits the highest potential score defines the type of mistake that has been made. That is, only ever one mistake is said to have been made, even if the highest scores from more than one type of available contract exceeded the current score. The unweighted mistakes of each type are stored in the variables *defender_missed_higher_contract*, *defender_missed_current_double*, and *defender_erroneously_doubled*. The weighted values, computed as the difference between the highest score achievable and the current score, are stored in the variables *weighted_defender_missed_higher_contract*, *weighted_defender_missed_current_double*, and *weighted_defender_erroneously_doubled*.

If the partnership being evaluated is the declaring side in the current contract, there are six types of contracts that could have been available to them. The procedure is the same as that for analysing the defender's play. The largest possible score from each type of available contract is considered, and should the maximum value among these scores exceed the current score, the variables for the type of mistake associated with the available contract that admitted this maximum score are increased. The first type of available contract to the declarers is any contract higher than the defender's last contract bid. If the defenders never made a contract bid, every contract is available. Again, any contract that the declarers could not make would be assumed to be doubled by the opponents. And again, perhaps only one of the players in the partnership could be the declarer for particular suits. If the defenders had made a contract bid at some point, the second type of available contract would be doubling the defender's last bid. Third, if the defender's last bid was erroneously redoubled by them, this could also have been considered an available contract. Fourth, any previous contract bid made by the declarers since the defenders' last bid that was erroneously doubled by the defenders, which may include the current contract, could have been redoubled by the declarers. Fifth, if the defenders had made a contract bid at some point that they would make, the declarers could have accepted this contract as final. Finally, if the declarers redoubled the current contract, not having redoubled it was an available type of contract to them as well. The six unweighted variables that will hold the number of times each of these types of mistakes were made are *declarer_missed_higher_contract*, *declarer_missed_doubling_opponents_last_bid*, *declarer_missed_opponents_last_redouble*, *declarer_missed_redoubling_opponents_erroneous_double*, *declarer_missed_opponents_last_bid* and *declarer_erroneous_redouble*. The weighted equivalents are *weighted_declarer_missed_doubling_opponents_last_bid*, *weighted_declarer_missed_opponents_last_redouble*, *weighted_declarer_missed_redoubling_opponents_erroneous_double*, *weighted_declarer_missed_opponents_last_bid* and *weighted_declarer_erroneous_redouble*.

When determining which auction mistake was made by the defenders or the declarers, it is possible that more than one mistake type had the same magnitude, and that this mistake magnitude was larger than or equal to all other mistake magnitudes. In this case, the values that would have been added to the relevant variables are divided by the number of mistake types involved in the tie. This way, the sum of all unweighted mistake variables still only increased by 1, and the sum of all weighted mistake variables still only increased by the magnitude of the largest mistake.

A summary of the thirty-three variables collected for each player by the double dummy analysis program is provided in Table 3. Together, these variables judge a player on every possible aspect of their game.

Table 3

Variables collected by the double dummy analysis program

Variable	Description
claims	The number of boards the player was involved in which ended in a claim.
claim_mistakes	The number of times the player was involved in a claim that awarded them fewer tricks than could have been won from optimal play.
weighted_claim_mistakes	The total score jeopardized by making claim mistakes.
leads	The number of times the player led to the first trick.
lead_mistakes	The number of times the lead was suboptimal.
weighted_lead_mistakes	The total score jeopardized by making lead mistakes.
cards_played_as_declarer	The number of cards played while the declarer.
play_mistakes_as_declarer	The number of suboptimal cards played as declarer.
weighted_play_mistakes_as_declarer	The total score jeopardized by making play mistakes while the declarer.
cards_played_as_defender	The number of cards played while a defender.
play_mistakes_as_defender	The number of suboptimal cards played as defender.
weighted_play_mistakes_as_defender	The total score jeopardized by making play mistakes while a defender.
auctions_analyzed	The number of auctions a player was involved in.
defender_missed_current_double	The number of times that the best available contract for the player while a defender was to double the current contract.
weighted_defender_missed_current_double	The total score jeopardised by not doubling the current contract when doing so was the best option.
defender_erroneously_doubled	The number of times that the best available contract for the player while a defender was to not double the current contract, yet they did.

Table 3 (Continued)

weighted_defender_erroneously_doubled	The total score jeopardised by doubling the current contract when the best option was not to have.
defender_missed_higher_conctract	The number of times that the best available contract for the player while a defender was one higher than the current contract.
weighted_defender_missed_higher_conctract	The total score jeopardised by not entering the best higher contract while a defender, when it was the best available contract.
declarer_missed_higher_contract	The number of times that the best available contract for the player while a declarer was one higher than the current contract.
weighted_declarer_missed_higher_contract	The total score jeopardised by not entering the best higher contract while a declarer, when it was the best available contract.
declarer_missed_doubling_opponents_last_bid	The number of times that the best available contract for the player while a declarer was to double the opponent's last bid.
weighted_declarer_missed_doubling_ opponents_last_bid	The total score jeopardised by not doubling the opponents' last bid when doing so was the best option.
declarer_missed_opponents_last_redouble	The number of times that the best available contract for the player while a declarer was to stay at the opponents' last contract which they redoubled.
weighted_declarer_missed_opponents_ last_redouble	The total score jeopardised by not staying at the opponent's last redoubled contract when doing so was the best option.
declarer_missed_redoubling_opponents_ erroneous_double	The number of times that the best available contract for the player while a declarer was to redouble an opponent's erroneous double on a previous contract that was higher than the defender's last bid.
weighted_declarer_missed_redoubling_ opponents_erroneous_double	The total score jeopardised by not redoubling the best opponent's erroneous double when doing so was the best option.

Table 3 (Continued)

declarer_missed_opponents_last_bid	The number of times that the best available contract for the player while a declarer was to stay at the opponents' last bid, not doubling it.
weighted_declarer_missed_opponents_last_bid	The total score jeopardised by not staying at the opponent's last contract when doing so was the best option.
declarer_erroneous_redouble	The number of times that the best available contract for the player while a declarer was to not redouble the current contract, yet they did.
weighted_declarer_erroneous_redouble	The total score jeopardised by redoubling the current contract when the best option was not to have.
passed_higher_contract	The number of times the player was involved in a board that was passed out, despite their partnership being able to make a contract.
weighted_passed_higher_contract	The total score jeopardised by not entering the best contract that could have been made.

The implementation of the double dummy analysis program is here.

The variables from Table 3 can then be used to compute fourteen weighted and fourteen unweighted variables that count a player's mistakes of each type whilst considering the number of opportunities they had to make such mistakes. These variables serve as averages, and their formulas are given in Table 4.

Table 4

Variables to be used in multiple linear regression analysis

Variable	Formula
claim_mistakes_per_claim	*claim_mistakes / claims*
weighted_claim_mistakes_per_claim	*weighted_claim_mistakes / claims*
lead _mistakes_per_lead	*lead_mistakes / leads*
weighted_lead_mistakes_per_lead	*weighted_lead_mistakes / leads*
defender_mistakes_per_card_played_as_ defender	*play_mistakes_as_defender / cards_played_as_defender*
weighted_defender_mistakes_per_card_ played_as_defender	*weighted_play_mistakes_as_defender / cards_played_as_defender*
declarer_mistakes_per_card_played_as_ declarer	*play_mistakes_as_declarer / cards_played_as_declarer*
weighted_declarer_mistakes_per_card_ played_as_declarer	*weighted_play_mistakes_as_declarer / cards_played_as_declarer*
defender_missed_current_double_per_ auction	*defender_missed_current_double / auctions_analyzed*
weighted_defender_missed_current_double _per_auction	*weighted_defender_missed_current_double / auctions_analyzed*
defender_erroneously_doubled_per_ auction	*defender_erroneously_doubled / auctions_analyzed*
weighted_defender_erroneously_doubled_ per_auction	*weighted_defender_erroneously_doubled / auctions_analyzed*
defender_missed_higher_conctract_per_ auction	*defender_missed_higher_conctract / auctions_analyzed*
weighted_defender_missed_higher_ conctract_per_auction	*weighted_defender_missed_higher_conctract / auctions_analyzed*
declarer_missed_higher_contract_per_ auction	*declarer_missed_higher_contract / auctions_analyzed*
weighted_declarer_missed_higher_ contract_per_auction	*weighted_declarer_missed_higher_contract / auctions_analyzed*

Table 4 (Continued)

declarer_missed_doubling_opponents_last_bid_per_auction	*declarer_missed_doubling_opponents_last_bid / auctions_analyzed*
weighted_declarer_missed_doubling_opponents_last_bid_per_auction	*weighted_declarer_missed_doubling_opponents_last_bid / auctions_analyzed*
declarer_missed_opponents_last_redouble_per_auction	*declarer_missed_opponents_last_redouble / auctions_analyzed*
weighted_declarer_missed_opponents_last_redouble_per_auction	*weighted_declarer_missed_opponents_last_redouble / auctions_analyzed*
declarer_missed_redoubling_opponents_erroneous_double_per_auction	*declarer_missed_redoubling_opponents_erroneous_double / auctions_analyzed*
weighted_declarer_missed_redoubling_opponents_erroneous_double_per_auction	*weighted_declarer_missed_redoubling_opponents_erroneous_double / auctions_analyzed*
declarer_missed_opponents_last_bid_per_auction	*declarer_missed_opponents_last_bid / auctions_analyzed*
weighted_declarer_missed_opponents_last_bid_per_auction	*weighted_declarer_missed_opponents_last_bid / auctions_analyzed*
declarer_erroneous_redouble_per_auction	*declarer_erroneous_redouble / auctions_analyzed*
weighted_declarer_erroneous_redouble_per_auction	*weighted_declarer_erroneous_redouble / auctions_analyzed*
passed_higher_contract_per_auction	*passed_higher_contract / auctions_analyzed*
weighted_passed_higher_contract_per_auction	*weighted_passed_higher_contract / auctions_analyzed*

The variables from Table 4 can then be used in the multiple linear regression analysis. Results and code files may use slightly different names for the variables from Table 4.

2.3. Multiple Linear Regression

Either the fourteen weighted or the fourteen unweighted variables from Table 4 can now be used as the set of independent variables that could predict a player's ability. The dependent variable will be the player's current world ranking. As discussed in the introduction, there are ten distinct ranking systems that could be used. The open masterpoints and open placing points will each be used separately. This then admits four datasets that will be analysed, each combination of the weighted and unweighted variables for the independent variables, and the masterpoints and placing points for the dependent variable.

Each of the independent variables defines an amount of mistakes. As such, one would expect each variable independently, and thus all variables when used together, to be negatively correlated with the dependent variable. Furthermore, one might expect the relationship to be linear, as the quality of a player will get progressively worse as the number of mistakes they make increases, and likely at a constant rate. Thus, multiple linear regression can be used in an attempt to find a correlation between the number of mistakes made and the player's ability.

To ensure meaningful results, players must only be included as data points if they had a large number of opportunities to make each type of mistake. That is, the five variables used as the denominators in Table 4 must be sufficiently large. For each of the independent variables in each data point to be meaningful, a player must meet all of these requirements simultaneously. The thresholds must be chosen large enough to ensure that each value is meaningful, yet small enough to still allow a significant number of data points. The thresholds will be determined through trial and error. Filtering out data points of players who do not meet these thresholds will then determine the four datasets that can be analysed using multiple linear regression.

To perform the multiple linear regression analysis, the statistical software package SAS will be used.

Before attempting to fit a model, each independent variable's variance inflation factor (VIF) will be computed. The variance inflation factor for a given independent variable j, denoted

VIF_j is computed as in Equation 1. In the equation, R_j is the coefficient of determination of the model that uses j as the dependent variable, and the remainder of the independent variables as the independent variables. If an independent variable has a large VIF value, typically larger than 5, it signifies that there is high multicollinearity between the independent variables. If an independent variable is highly correlated with the rest of the independent variables, there is no need to consider it as a predictor candidate in a regression model. Thus, variables with a large VIF value will be removed from consideration, and all of the VIF values will then be recomputed. This procedure may be continued until all remaining independent variables have acceptably low VIF values. This subset of independent variables can then be considered during the model selection procedure.

$$VIF_j = \frac{1}{1 - R_j^{2}} \qquad (1)$$

The model selection procedure can now be used to suggest potential models. Three different model selection procedures will be used on each of the four datasets, for a total of twelve suggested models. The three procedures are forward selection, backward elimination, and stepwise regression. The forward selection technique begins with an empty model. It then introduces the most significant predictor into the model. It continues this process until there are no more variables that are significant at the specified alpha-entry level. SAS uses a lenient default alpha-entry value of 0.5 for forward selection, which will be left unchanged for this analysis. The backward elimination model begins with the full model containing all variables. It then continuously removes the least significant predictor from the model one variable at a time, until all variables in the model are significant at the alpha-stay level. SAS uses a default alpha-stay value of 0.1 for backward selection, and this will also remain unchanged for this analysis. Finally, stepwise regression combines the methodology of the other two procedures. It begins with an empty model, as does forward selection. Also like forward selection, stepwise regression introduces the most significant variables into the model, one at a time. If at any point, the significance of a predictor in the model drops below the alpha-stay level, it will be removed from the model. The procedure is finished when every variable in the model is significant at the alpha-stay level, and every variable not in the model does not meet the alpha-entry requirement. Or, if the variable to be introduced is the one that was just removed. SAS uses both an alpha-stay and alpha-entry value of 0.15, and this analysis will not modify either value.

The adjusted coefficient of determination will then be computed for each model to aid in the decision-making process to follow. By considering the adjusted coefficients of determination, the p-value of each model, the p-value of each predictor inside each model, and the coefficients the model suggests for each predictor, the most appropriate model will be chosen subjectively. Knowledge of the domain prohibits including predictors in the model with positive coefficients, as it is known that every independent variable is expected to be negatively correlated with the response variable. The inclusion of predictors with positive coefficients is a result of overfitting, and they make the model less generalizable and thus less useful. There is also a preference for simpler models. That is, if adding many more predictors only improves the coefficient of determination slightly, the model with fewer predictors may be preferred.

Once a model is selected, it should be tested to ensure that it satisfies the assumptions of linear regression. If it does not, the predictions made by the model will not be as accurate as the correlation implies. The model must be tested to ensure that the error terms are normally distributed with a mean of zero and equal variance and that the relationship between the predictors and the response variable is indeed linear. To do so, two plots will be produced. Firstly, a plot of residuals versus predicted values will be studied. The plot should show an even, random scatter of points above and below zero. This would confirm linearity and that the error terms have a mean of zero and a constant variance. To test if the error terms are normally distributed, a qq-plot will be considered. One should see a relatively straight line formed, without excessive curvature. The interpretation of these graphs is largely subjective.

The model found by this research can then be used to determine a ranking system for players of the card game of bridge. Whether this ranking system should actually be used will depend on the strength of the model, and whether or not it satisfies the assumptions of linear regression.

3. Results

Firstly, the data cleaning program was run on each board in the database. The output of the program, which will serve as the input to the double dummy analysis program, is here. This file is evaluable as a Python object. The number of boards that were excluded, and the reasoning as to why, is outlined in Table 5.

Table 5

Frequency of reasons for board exclusion

Reason for exclusion	Number of boards excluded
declarer differs from the declarer deducible from the auction.	7757
The declarer claimed fewer tricks than they had already won, or more tricks than there were remaining.	190
deal had a hand that did not have 13 cards.	18
All names are missing.	1386
contract differs from the contract deducible from the auction.	416
lead differs from the lead deducible from the declarer.	37
A player had an invalid card in the *deal*, or a card appeared twice.	11
Missing *deal*.	34
contract is "P" and *play* is non-empty.	1
Total	9850
No error	190622

Of the 190622 remaining boards, some had an invalid *auction* variable. The frequency of each reason for auction exclusion is summarised in Table 6. It shows that 190619 boards can have their auction analysed.

Table 6

Frequency of reasons for auction exclusion

Reason for *auction* exclusion	Number of boards with excluded *auction*
Missing *auction*.	2
The contract was passed out in the middle of the auction, or the auction did not end with three passes.	1
A contract bid was made in a non-increasing fashion.	0
A double bid was made where the contract was already doubled or redoubled, the bid was made by the side who bid the last contract, or there was no contract bid made yet.	0
A redouble bid was made where the contract was not doubled or already redoubled, the bid was made by the side who did not bid the last contract, or there was no contract bid made yet.	0
Total	3
No error	190619

Additionally, some of the boards had an invalid *play* variable. Table 7 summarises the reasons for the invalidity and how many boards had their *play* excluded for the given reason. It shows that 183761 boards can have their *play* analysed.

Table 7

Reasons for play exclusion

Reason for *play* exclusion	Number of boards with excluded *play*
Missing *play*, *lead*, or *contract*.	6001
Dashes appeared in more than one trick.	3
A card is played twice or the player never had it.	699
A player reneged.	139
A card is played to a trick after a dash.	6
First card played to a trick is a dash.	6
The number of legal plays, augmented with dashes if necessary, is not divisible by 4.	7
Total	6861
No error	183761

It is possible that some of the 190622 boards included had both an invalid *auction* and an invalid *play*. These boards, if they exist, will not get analysed in any way.

The double dummy analysis program was then run on the cleaned boards. The runtimes of the data cleaning and double dummy analysis programs are in Table 8.

Table 8

Data cleaning and Double Dummy Analysis Program Runtimes

Program	Runtime in Seconds
Data cleaning	3621.9906
Double Dummy Analysis	12678.0137

The results for every player in the database and the values of the thirty-three variables from Table 3 that define their game are here. The values of the averaged variables from Table 4 are split between the fourteen weighted mistake variables, which are here, and the fourteen unweighted mistake variables, which are here. The double dummy analysis results show that 2145 players from the database were able to be matched to a name from the WBF rankings, and were involved in at least one board.

Using trial and error to determine the minimum number of opportunities to make each type of mistake that a player must have had to be included as a data point in the regression analysis yielded the decisions described in Table 9.

Table 9

Minimum values required to be included as a data point

Variable	Minimum value required
claims	362
leads	100
cards_played_as_declarer	1408
cards_played_as_defender	1424
auctions_analyzed	392

Applying this filter reduced the number of players for which there are results from 2145 to 200. The weighted results for only these 200 players are here, and the unweighted results are here.

The VIF for each independent variable was then calculated. The results for the unweighted variables are in Table 10, and the results for the weighted variables are in Table 11.

Table 10

VIF values for each unweighted variable

Variable	VIF
claim_mistakes_per_claim	1.09473
lead_mistakes_per_lead	1.09965
declarer_mistakes_per_card_played_as_declarer	1.11604
defender_mistakes_per_card_played_as_defender	1.23361
defender_missed_current_double_per_auction	1.22138
defender_erroneously_doubled_per_auction	1.11691
defender_missed_higher_contract_per_auction	1.27829
declarer_missed_higher_contract_per_auction	1.24355
declarer_missed_doubling_opponents_last_bid_per_auction	1.16026
declarer_missed_opponents_last_redouble_per_auction	1.11373
declarer_missed_redoubling_opponents_erroneous_double_per_auction	1.11763
declarer_missed_opponents_last_bid_per_auction	1.18066
declarer_erroneous_redouble_per_auction	1.07692
passed_higher_contract_per_auction	1.20795

Table 11

VIF values for each weighted variable

Variable	VIF
weighted_claim_mistakes_per_claim	1.20808
weighted_lead_mistakes_per_lead	1.07679
weighted_declarer_mistakes_per_card_played_as_declarer	1.07398
weighted_defender_mistakes_per_card_played_as_defender	1.12582
weighted_defender_missed_current_double_per_auction	1.22832
weighted_defender_erroneously_doubled_per_auction	1.11364
weighted_defender_missed_higher_contract_per_auction	1.11538
weighted_declarer_missed_higher_contract_per_auction	1.04436
weighted_declarer_missed_doubling_opponents_last_bid_per_auction	1.09177
weighted_declarer_missed_opponents_last_redouble_per_auction	1.13641
weighted_declarer_missed_redoubling_opponents_erroneous_double_per_auction	1.07121
weighted_declarer_missed_opponents_last_bid_per_auction	1.21086
weighted_declarer_erroneous_redouble_per_auction	1.05677
weighted_passed_higher_contract_per_auction	1.12281

From Table 10 and Table 11, it is clear from the small VIF values that there is no multicollinearity between any of the independent variables. Thus, they may all be considered for inclusion in the multiple linear regression model.

Forward selection, backward elimination and stepwise regression model selection techniques were then run on each of the four data sets, and the adjusted coefficient of determination values were computed for the final model from each procedure. The SAS code used to produce the results for each dataset is here. The results are summarised in Table 12. The

Variable p-value column contains the p-value of that variable in the final model selected by the given procedure on the given dataset.

Table 12

Model selection results

Dataset	Procedure	Variables included	Variable p-value	Model p-value	Adj. R^2
MP, weighted	Stepwise	*weighted_defender_mistakes_per_ card_played_as_defender*	0.1139	0.1139	0.0076
MP, weighted	Backward	None	None	None	None
MP, weighted	Forward	*weighted_defender_mistakes_per_ card_played_as_defender*	0.1029	0.5187	-0.0041
		weighted_claim_mistakes_per_claim	0.3344		
		weighted_defender_missed_higher_ contract_per_auction	0.4637		
		weighted_declarer_missed_higher_ contract_per_auction	0.3365		
		weighted_lead_mistakes_per_lead	0.3590		
		weighted_defender_erroneously_ doubled_per_auction	0.4113		
		weighted_declarer_erroneous_ redouble_per_auction	0.4697		
		weighted_declarer_missed_ opponents_last_bid_per_auction	0.3528		
MP, unweighted	Stepwise	*defender_mistakes_per_card_played _as_defender*	0.0587	0.0027	0.0605
		claim_mistakes_per_claim	0.1450		
		defender_missed_higher_contract_ per_auction	0.0223		
		declarer_missed_opponents_last_ bid_per_auction	0.0495		

Table 12 (Continued)

MP, unweighted	Backward	*defender_mistakes_per_card_played_as_defender*	0.0302	0.0028	0.0550
		defender_missed_higher_contract_per_auction	0.0220		
		declarer_missed_opponents_last_bid_per_auction	0.0582		
MP, unweighted	Forward	*defender_mistakes_per_card_played_as_defender*	0.0628	0.0050	0.0585
		claim_mistakes_per_claim	0.1857		
		declarer_missed_opponents_last_bid_per_auction	0.0597		
		defender_missed_higher_contract_per_auction	0.0175		
		declarer_missed_higher_contract_per_auction	0.4431		
PP, weighted	Stepwise	*weighted_defender_mistakes_per_card_played_as_defender*	0.0004	0.0010	0.0579
		weighted_defender_missed_current_double_per_auction	0.1281		
PP, weighted	Backward	*weighted_defender_mistakes_per_card_played_as_defender*	0.0007	0.0007	0.0516

Table 12 (Continued)

PP, weighted	Forward	*weighted_defender_mistakes_per_card_played_as_defender*	0.0003	0.0095	0.0548
		weighted_defender_missed_current_double_per_auction	0.0802		
		weighted_claim_mistakes_per_claim	0.2995		
		weighted_declarer_missed_opponents_last_bid_per_auction	0.2543		
		weighted_passed_higher_contract_per_auction	0.4706		
		weighted_declarer_mistakes_per_card_played_as_declarer	0.2740		
PP, unweighted	Stepwise	*defender_mistakes_per_card_played_as_defender*	<0.0001	<0.0001	0.0934
		defender_missed_higher_contract_per_auction	0.1358		
		declarer_missed_opponents_last_bid_per_auction	0.0519		
PP, unweighted	Backward	*defender_mistakes_per_card_played_as_defender*	<0.0001	<0.0001	0.0877
		declarer_missed_opponents_last_bid_per_auction	0.0266		

Table 12 (Continued)

PP, unweighted	Forward	*defender_mistakes_per_card_played_as_defender*	0.0001	0.0003	0.0949
		defender_missed_current_double_per_auction	0.1263		
		declarer_missed_opponents_last_bid_per_auction	0.0297		
		claim_mistakes_per_claim	0.2854		
		declarer_mistakes_per_card_played_as_declarer	0.3890		
		defender_missed_higher_contract_per_auction	0.1702		
		declarer_missed_doubling_opponents_last_bid_per_auction	0.2714		

A model can now be subjectively chosen based on the results in Table 12. It is clear from the results that the unweighted independent variables were more correlated to the response variables than the weighted independent variables. This is best illustrated by considering the difference in the adjusted R^2 values between models that use the same response variable but a different set of independent variables. The same type of comparison can be used to show that placing points yielded a better correlation than masterpoints, by holding the independent variable set used constant and changing the response variable. Thus, the models selected by using the unweighted variables and the placing points will be the ones that are considered further. The model on this dataset with the highest adjusted coefficient of determination was the one determined by the forward selection procedure. The coefficients on each variable determined by this model are expressed in Table 13. The **p-value**s in Table 13 are the same as they were for this model in Table 12 and are simply repeated for convenience.

Table 13

Predictor coefficients for the model determined by forward selection using unweighted variables and placing points

Variable	Coefficient	P-value
Intercept	25.52042	0.2097
defender_mistakes_per_ card_played_as_defender	-1069.96683	0.0001
declarer_mistakes_per_ card_played_as_declarer	-262.60986	0.3890
claim_mistakes_per_claim	-149.34894	0.2854
defender_missed_current_double_per_auction	117.98612	0.1263
defender_missed_higher_contract_per_auction	-122.43156	0.1702
declarer_missed_doubling_opponents_last_bid _per_auction	116.57173	0.2714
declarer_missed_opponents_last_bid_per_ auction	520.189555	0.0297

As each predictor is expected to be negatively correlated with the response variable, the many positive coefficients in the model described in Table 13 are prohibited by knowledge of the domain. The resulting adjusted R squared of 0.0949 for the model is thus inflated by these overfittings, and this model should not be used as it will not be generalizable to other data.

The other two models obtained from using the unweighted variables and the placing points had the second and third highest values of adjusted R^2, respectively. The coefficients on the predictors for these models are shown in Table 14 and Table 15, respectively. Again, the p-values are repeated for convenience.

Table 14

Predictor coefficients for the model determined by stepwise selection using unweighted variables and placing points

Variable	Coefficient	P-value
Intercept	42.14957	<0.0001
defender_mistakes_per_card_played_as_defender	-1123.59216	<0.0001
defender_missed_higher_contract_per_auction	-129.09030	0.1358
declarer_missed_opponents_last_bid_per_auction	460.53935	0.0519

Table 15

Predictor coefficients for the model determined by backward elimination using unweighted variables and placing points

Variable	Coefficient	P-value
Intercept	31.74272	<0.0001
defender_mistakes_per_card_played_as_defender	-1167.15377	<0.0001
declarer_missed_opponents_last_bid_per_auction	520.07566	0.0266

The smaller models described in Table 14 and Table 15 are more promising, as most of the predictors with significant positive coefficients, and less significant predictors with negative coefficients, have been removed. However, the significant positively correlated predictor *declarer_missed_opponents_last_bid_per_auction* remains in each model, and thus they should not be used.

In fact, the only predictor that was fitted with a negative coefficient that was significant at the typical 0.05 alpha level, was *defender_mistakes_per_card_played_as_defender*. This predictor was very significant in all three highlighted models, with a p-value of less than or equal to 0.0001 in each model. As such, the simple linear regression model consisting of only this predictor is considered. The results are in Table 16.

Table 16

Predictor coefficient for the model consisting only of defender_mistakes_per_card_played_as_defender

Variable	Coefficient	P-value
Intercept	35.43033	<0.0001
defender_mistakes_per_card_played_as_defender	-1028.60322	<0.0001

The adjusted R^2 for the model described in Table 16 was found to be 0.0693. This model is chosen as the best model as it is simple, significant, and does not include any predictors with positive coefficients.

Now, the assumptions of linear regression are tested on this model to gain more insight as to whether or not it can be used. First, the plot of residuals versus predicted value is considered, and this is shown in Figure 1.

Figure 1

Plot of residuals versus predicted value

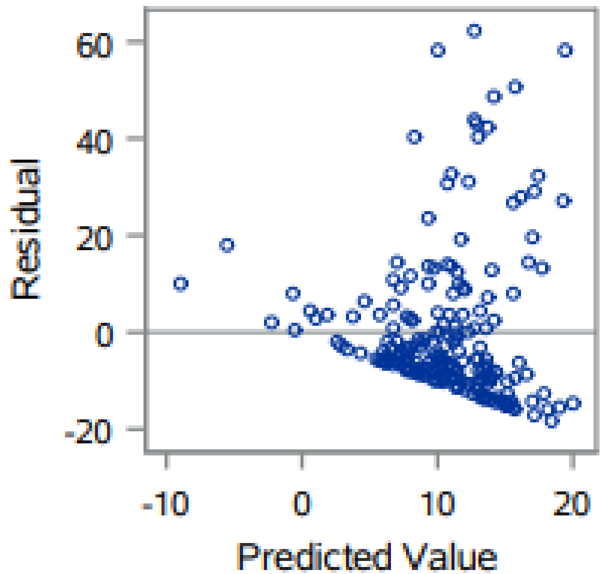
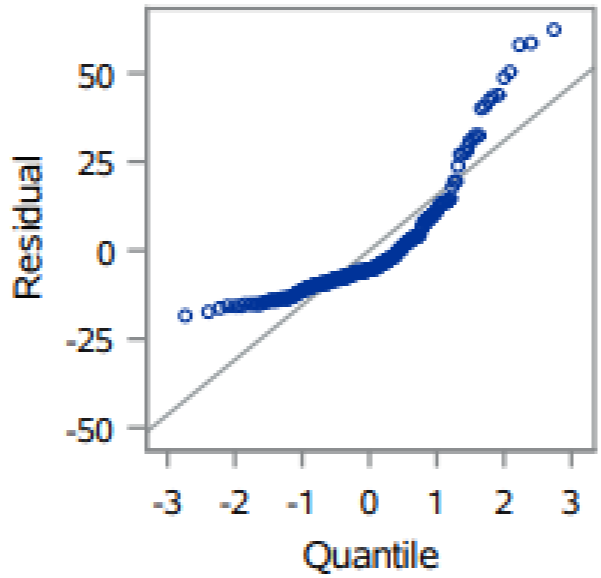


Figure 1 shows the range of residuals widen as predicted value increases. This suggests that the error terms are heteroscedastic, and thus the assumption of their constant variance is violated.

Secondly, the qq-plot is shown in Figure 2.

Figure 2

qq-plot



The points in Figure 2 do deviate from the diagonal guideline, but not overly so. The pattern formed does appear to have relatively strong curvature. Although it can not be said with certainty that the assumption is violated, it does appear as if the distribution of the error terms differs slightly from that of a normal distribution.

Overall, this model states that the only significant predictor of a player's ability is the percentage of time they make a mistake while playing a card as a defender. The ranking system admitted by this model is then described by Equation 2. Here, x is set equal to the value of a player's *defender_mistakes_per_card_played_as_defender* variable as computed by the double dummy analysis program. The y value is then equal to the number of placing points the player would be expected to have, given their play. Players can then be ranked by this predicted number of placing points.

$$y = -1028.60322x + 35.43033 \tag{2}$$

4. Discussion

The model developed in this research should not be trusted to rank bridge players accurately. Although the model is determined significant by its low p-value, the adjusted R^2 analysis and the assumption tests reveal a weak model. The adjusted R^2 value of only 0.0693 indicates that the model only describes 6.93% of the variability in the placing points. So, while it is the case that an increasing value for *defender_mistakes_per_card_played_as_defender* generally leads to a lower number of placing points as expected, the variability is much too high to make accurate predictions. In addition to this shortcoming, the intercept was found to be only 35.43033. Thus, a player who makes no mistakes as a defender whatsoever would be predicted to have 35.43033 placing points. Comparing this to the current world rankings by open placing points reveals that a player of this quality would only be ranked 23rd in the world. Finally, the assumption tests clearly showed a violation of the condition that the error terms must have constant variance.

The result that the placing points yielded a higher correlation than the masterpoints was expected. The boards from the database were played in years between 1955 and 2016. So, many talented players who achieved low scores for their mistake variables in these boards may have retired a significant amount of time ago. As such, their masterpoints could have decayed to a very small amount or even zero. However, their placing points would have remained throughout time. The result that unweighted variables are better at predicting a player's ability than weighted variables is a more surprising one. One would expect the magnitude of a player's mistakes to influence their ranking significantly. However, this result shows that the weighted average only adds noise to the data, and it is instead the frequency of the mistakes that is significant. Despite the inability to determine an accurate model in this research, one would still expect the values of the collected variables, weighted or unweighted, to be an accurate measure of determining a player's ability. Furthermore, one would trust the world ranking systems to give out ranking points based on a player's ability. So, there must have been significant enough biases introduced in the data used or in the methodology employed to result in the variables not being good predictors of a player's ability.

One potential source of bias is in the world rankings themselves. The rankings by placing points do not account for the change in the relative ability of players across large time periods. To illustrate this point, consider the possibility that the average ability of a bridge player who competes in a prestigious tournament is much higher today than it was in the past. The victors of the previous tournaments may have received the same number of placing points as the victors of today, despite being lesser players. In addition to this, the rankings by placing points will naturally be biassed towards players who have been playing for longer. A younger player, who has demonstrated their ability at the highest level, may still be ranked lower than older, lesser players who have entered more tournaments and thus had more of a chance to collect placing points.

Another main source of potential bias is in the dataset used. The data may have been heavily focused on certain tournaments or time periods. There may not have been an adequate spread of data that allowed players from all levels of the rankings to be represented in the regression. Furthermore, there may have been boards in the dataset that were constructed in a way that were technically legal bridge boards, but that contained clearly fabricated information. For example, during a manual inspection of the database, several boards were discovered that listed the deal as each player having every card of a particular suit. There was also an improbable number of boards that stated a claim occurred after only the first card. Obviously, applying the same analysis in this research on a different dataset could give dramatically different results.

The method of comparing a player's play to that of the DDS could also introduce some bias. The DDS uses full information of all four hands to make its decisions. The players must operate on limited information. Thus, there may be a scenario in which the player made the best decision given the information available, but the DDS still classifies their play as a mistake. Although, this does not invalidate the results as all players are subjected to the same analysis, and better players will still perform closer to how the DDS performs than lesser players will. It does however mean that a player playing perfectly may not receive a value of zero for all mistake variables.

Other biases in the double dummy analysis methodology could arise from how the *year* or *vulnerability* variables are computed. The scoring of boards depends both on the year and whether the declaring side is vulnerable. If the *year* variable was unable to be populated, it defaulted to a value that led to the most recent scoring system being used. If the board was truly played using the other scoring system, the mistake variables may be incorrectly updated. The reason that boards for which the year could not be determined are still analysed and not discarded, is that only a small percentage of the boards were able to have their *year* variable populated. Misidentifying the *vulnerability* could also lead to incorrect analysis. As there were only 361 boards in the database for which *vulnerability* could not be determined, it was intended that these boards were to be discarded and not analysed. Although, an oversight in the program led to the *vulnerability* being set to not vulnerable for both sides when *vulnerability* was empty.

Another potential source of bias results from the imprecise fuzzy matching algorithm that had to be used to match names from the database to names from the world rankings. Although the algorithm was observed performing well, it cannot be guaranteed that the algorithm did not make a mistake. Thus, a player from the world rankings may be attributed mistakes that they did not commit.

The process of choosing the thresholds for the five variables that defined the number of opportunities a player had to make each type of mistake, could also have introduced bias. By restricting the dataset to players who appeared many times in the database, the scope of the data could become narrower. To account for this, these thresholds could have been modified throughout the regression model selection procedure. If, for example, it was determined that mistakes relating to claims would not be significant in any model, the restriction on the *claims* variable could have been removed to allow for more data in the remaining models.

The final known source of potential bias stems from different player categories defined by the WBF and the different ranking systems used. The regression analysis only attempted using the open masterpoints and open placing points as response variables. However, there are boards in the database that were from tournaments that would award different types of points. For example, players from the database who performed very well, but only played in youth tournaments, are not going to see a correlation between their mistake variables and their open masterpoint or placing point ranking, as they may not have any open points at all.

Taking measures to reduce the biases discussed could lead to better results.

5. Conclusion

In conclusion, this research attempted, but failed, to provide a new ranking system for players of the card game of bridge based on their hand records. 190622 cleaned boards were extracted from the original data file. Of these, 190619 boards were able to have their play portion analysed, and 183761 boards were able to have their auction analysed. The double dummy analysis program run on these boards provided the values of the weighted and unweighted mistake variables for use in the regression analysis. The process of removing players from the list of data points, if they had not participated in a sufficient number of scenarios, concluded with 200 players to serve as data points. Each of these 200 players was involved in at least 362 claims, 100 leads, and 392 auctions, and played at least 1408 cards as the declarer and at least 1424 cards as a defender. The VIF was low for all collected variables, so they were all included in the regression model selection procedure. It was found that using placing points as the response variable yielded a higher correlation than using masterpoints. Furthermore, the unweighted mistakes variables were better at predicting a player's ability than the weighted variables. The final model consisted only of the *defender_mistakes_per_card_played_as_defender* variable with a coefficient of -1028.60322 and an intercept of 35.43033. The model had an R^2 value of 0.0693 and was found to violate the assumption of a constant variance between error terms. As such, the model was rejected and the attempt to determine a new accurate ranking system failed. Although, the expectation that the variables collected in this research should completely define a player's ability is not lost. Thus, future works are encouraged to use the methodology and results from this research to continue the search for a new ranking system for players of the card game of bridge.

References

[1] American Contract Bridge League. (2017). Laws of Duplicate Bridge. https://web2.acbl.org/documentlibrary/play/laws-of-duplicate-bridge.pdf

[2] Haglund, B. (2018). Double Dummy Solver. https://privat.bahnhof.se/wb758135/bridge/

[3] Mendenhall, W., Beaver, R., Beaver, B., Ahmed, S. (2013). Introduction to Probability and Statistics (Third Canadian Edition). Nelson College Indigenous.

[4] World Bridge Federation. (2023). WBF Masterpoints. http://www.wbfmasterpoints.com/

Appendix A: The Rules of Bridge

Bridge is a trick-taking card game played with a standard 52-card deck. The game is played in several forms, most commonly divided into rubber bridge and duplicate bridge. Rubber bridge is used in recreational play, whereas duplicate bridge is used for competitive play. Here, we focus on the rules of duplicate bridge. The game is made up of **boards**. A board is contested by two partnerships of two players each. The players are called North, East, South, and West, and they sit in a circular fashion in that order. North and South form a partnership, while East and West form the other partnership. For a given board, each player is dealt 13 cards, so that every card from the deck is in play. For now, the cards of each player are known to that player only. One player is assigned to be the **dealer**. Either one partnership, both partnerships, or neither partnership is said to be **vulnerable**.

A board begins with the **auction.** Starting with the dealer, and rotating in a clockwise direction, players can make a **bid**. Players are forbidden to exchange any information through any other medium other than the bids they make. If a partnership has agreed upon special meanings for their bids, these meanings must be described precisely to the opponents before the board is played. Therefore, any information shared in the auction is revealed equally to all players. With regards to the bids players may make, they may either pass, make a contract bid, make a double bid, or make a redouble bid. Passing represents the absence of a bid, and the auction ends if three consecutive players pass following a non-pass bid, or if all four players pass without a non-pass bid having ever been made. In the latter case, the auction is said to have been passed out. A contract bid is a bid that consists of a **level** and a **trump** suit. The levels are between 1 and 7 inclusive, while the possible trump suits are clubs, diamonds, hearts, spades, and no trump. Thus, there are 35 possible contract bids. Contract bids are ordered such that a higher level corresponds to a larger bid, and within a level, the bids are ordered by trump suit in the aforementioned order. Once a contract bid is made, any subsequent contract bid must be of a higher ranking. A double bid can be made by a player if the most recent non-pass bid is a contract bid made by the opposing partnership. A player can make a redouble bid if the most recent non-pass bid is a double bid made by the opposing partnership. Once the auction ends, the **contract** is assigned to be the latest contract bid to have been made, alongside the indication of whether the contract bid was doubled, redoubled, or neither. The partnership that bid the eventual contract is said to have won the auction. Of the players belonging to the partnership that won the auction, the first player to have made a contract bid whose trump suit is equal to the trump suit of the eventual contract is said to be the **declarer**. The partner of the declarer is said to be the **dummy**. The players from the opposing partnership are called the **defenders**.

Following the auction, the second phase of a board begins, the **play**. The play begins with the defender to the left of the declarer, called the **lead**. The lead player may play any card they wish. Following the lead, the dummy reveals their cards for all players to see. The play then continues in a clockwise manner. When it is the dummy's turn, the declarer chooses the card that will be played from the dummy's hand. If a player has a card of the suit that is lead, they must play one of them. If they fail to do this, it is called reneging and is against the laws of the game. If they do not have any cards with the same suit as the card lead, they may play any card. The act of each player playing one card in this manner is called a **trick**. Once all four players have played a card, the trick is complete and the winner of the trick may be determined. To determine the winner of the trick, the rank and suit of each card played are considered. The ranks of the cards are ordered in the typical fashion, with ace being the highest and 2 being the lowest. The winner of the trick is the player who played the card of the trump suit with the highest rank. If no card played in the trick is of the trump suit, then the card with the highest rank that is of the lead suit wins the trick. If the contract is a no trump contract, then only the latter criterion is used. If a player wins a trick, then they lead the subsequent trick. Play continues in this manner until one of two things happens. Either, 13 tricks will be played and all players will be devoid of cards. Or, a player will make a **claim**. The defenders, or the declarer from their own position or from the dummy's position, may make a claim at any point in the play, even in the middle of a trick. A player makes a claim by stating how many tricks they believe they can take from the remainder of the play. If all players agree to the claim, the play ends and each side is said to have won a given amount of tricks. The number of tricks that each side wins is equal to the sum of the number of tricks they won in play, and the number of tricks that the claim specified they would go on to win. If no claim is made, the amount of tricks each side won is assigned to be the number of tricks they won from the play. If the declarer's side won at least 6 tricks more than the level of the contract, they are said to have made their contract. The first six tricks are called the

book. The tricks won beyond the book, up to the level of the contract, are called **odd tricks**. Any tricks that were won beyond the book and beyond the odd tricks are called **overtricks**. If the declarer's side did not win enough tricks to have made their contract, the difference between the number of tricks they did win and the number of tricks they would have needed to win to make their contract, are called **undertricks.**

Once the play has concluded, each side may be scored on their performance on the board. Scoring of a bridge board is a zero-sum game, meaning that the absolute value of the scores of the two sides will be the same, and one will be the additive inverse of the other. If the declarer's side makes the contract, they will earn a positive score. The magnitude of the score they earn is given by the sum of the **contract points**, **partial-game bonus**, **game bonus**, **overtrick points**, **slam bonus**, and **double bonus**.

Contract points are awarded for each odd trick taken. They depend on the trump suit, and whether the contract was undoubled, doubled, or redoubled. The values are summarised in Table A1.

Table A1

Contract points awarded per odd trick

Trump suit	Undoubled	Doubled	Redoubled
No trump, first trick	40	80	160
No trump, subsequent tricks	30	60	120
Spades or hearts	30	60	120
Clubs or diamonds	20	40	80

If the declarer's side makes the contract, they will be awarded some number of bonus points. The amount depends on how many contract points they earned from completing the contract, and their vulnerability. Earning fewer than 100 contract points awards a partial-game bonus, while scoring 100 or more contract points awards a game bonus. The values for these bonuses are outlined in Table A2.

Table A2

Contract points awarded per trick

Contract Points	Not vulnerable	Vulnerable
Fewer than 100	50	50
100 or more	300	500

Overtrick points are awarded for each overtrick. If the contract was not doubled or redoubled, the points awarded for each overtrick are exactly the same as they are in Table A1. The first overtrick for a no trump contract is scored as a regular trick, not as the first trick. The points awarded per overtrick when the contract is not undoubled are available in Table A3.

Table A3

Overtrick points awarded per trick

Declarer’s vulnerability	Doubled	Redoubled
Not vulnerable	100	200
Vulnerable	200	400

The declarer’s side will earn a small slam bonus if they bid a contract of level 6 and take at least 12 tricks. The declarer’s side will earn a grand slam bonus if they bid a contract of level 7 and take all 13 tricks. The amount of these bonuses depends on the declarer's vulnerability, and are in Table A4.

Table A4

Slam bonuses

Odd tricks	Not vulnerable	Vulnerable
6	500	750
7	1000	1500

If the contract was doubled or redoubled, the declarer earns an insult bonus. The values are shown in Table A5.

Table A5

Insult bonuses

Doubled	Redoubled
50	100

If the declarer did not make the contract, they receive a negative score equal to the number of **penalty points**. The defending side will then earn a positive score of the same magnitude. The number of penalty points awarded in each scenario is outlined in Table A6.

Table A6

Penalty points post 1987

Undertricks	Not vulnerable			Vulnerable		
	Undoubled	Doubled	Redoubled	Undoubled	Doubled	Redoubled
1st	50	100	200	100	200	400
2nd, 3rd	50	200	400	100	300	600
4th, 5th, …	50	300	600	100	300	600

The values from Table A6 have been in place for all years following 1987 until the time of writing. Prior to and in 1987, a not vulnerable doubled or redoubled contract would award a different number of penalty points for the 4th and subsequent undertricks. The penalty points table that was used in and before 1987 is shown in Table A7, with the modified values highlighted.

Table A7

Penalty points prior to and in 1987

Undertricks	Not vulnerable			Vulnerable		
	Undoubled	Doubled	Redoubled	Undoubled	Doubled	Redoubled
1st	50	100	200	100	200	400
2nd, 3rd	50	200	400	100	300	600
4th, 5th, …	50	200	400	100	300	600

There were other periods of time before 1987 that made use of slightly different rulesets, but they need not be mentioned here.

Once each side has been assigned a score based on the outcome of the board, bridge tournaments typically compare the score achieved by a partnership to the score achieved by one or more different partnerships who were sitting in the exact same position and playing the exact same board against a different set of opponents. This way, the element of luck is removed from the game. There are several different methods for converting the differences in the score earned to a scale on which players can be ranked, but they need not be discussed here.