Final Analysis 520

Gracie Inman

2023-08-12

Growing up it is a common dream to become rich and famous. Some people try to become famous by acting or singing. Some people try to become rich by inventing or investing. In my family we dreamt of hitting the lottery and becoming so rich we could buy our own island. Each state has their own lottery system as well as some spanning the nation. The lottery is said to be random, and anyone can win the huge jackpots. However, is it as random as they say? Is there a pattern to the numbers? Can I use the data put forth by the New York State Lottery to find a number combination that is more likely to win than the others?

To address this, I performed an analysis on the data from five different lottery games in the state of New York. The games are Powerball, Mega Millions, Cash 4 Life, Take 5, and the New York Lotto. I reviewed that data and cleaned it by removing incomplete data and empty columns. I began the analysis by just looking at the numbers individually and combined from each data set. I went on to run summary statistics on each of the data sets all data sets. While also performing a linear regression on each data set individually as well as together. For perspective I calculated the probability of winning the Powerball on any normal day. I finished by counting the number of times each combination of numbers appeared in each data set.

While Powerball did appear to have a larger number of smaller numbers and a small left skew. This also appeared in the graph of all the combined winning numbers. The data was otherwise normally distributed. I went on to run summary statistics on each of the data sets all data sets had a large standard deviation and p values of one suggesting no difference except for chance. The regression analysis showed significance and the residual standard error showed a good fit for each model. The probability of winning the Powerball on a normal day is 1.067757e-05, which are not good odds. I then calculated how many times each specific combination of numbers (in no order since they can be in any order to win) where drawn. To my surprise, most number combinations were only drawn once or twice.

According to my analysis, the lottery is random, and no significant patterns were identified in the data. The skew in numbers could be due to the different scales in each out of the lottery games. For example, Mega Millions has numbers up to 75 and Take 5 only contains numbers up to 39. This can lead to error when combining data sets due to the ranges being significantly different. However, looking at the data separately did not yield any significant conclusions either. All the p values were 1 meaning the only difference was due to chance in the data set. Since the data only consisted of numbers, it would be difficult to provide any additional conclusions. Having additional eyes and ideas may lead to additional opinions and methods of testing. However, I do not believe the ending result will change. Analysis of lottery data only confirmed that the lottery is in fact random, and I have the same chance of hitting the jackpot as anyone else (providing they only buy one ticket).

Data:

1. Publisher State of New York. (2023a, July 28). *Lottery cash 4 life winning numbers: Beginning 2014*. Catalog. https://catalog.data.gov/dataset/lottery-cash-4-life-winning-numbers-beginning-2014
2. Publisher State of New York. (2023, July 28). *Lottery mega millions winning numbers: Beginning 2002*. Catalog. https://catalog.data.gov/dataset/lottery-mega-millions-winning-numbers-beginning-2002
3. Publisher State of New York. (2023b, July 28). *Lottery NY Lotto winning numbers: Beginning 2001*. Catalog. https://catalog.data.gov/dataset/lottery-ny-lotto-winning-numbers-beginning-2001
4. Publisher State of New York. (2023, July 28). *Lottery Powerball winning numbers: Beginning 2010*. Catalog. https://catalog.data.gov/dataset/lottery-powerball-winning-numbers-beginning-2010
5. Publisher State of New York. (2023e, July 28). *Lottery take 5 winning numbers*. Catalog. https://catalog.data.gov/dataset/lottery-take-5-winning-numbers

Final Project Winning the Lottery

Gracie Inman

2023-08-12

Step One

library(pandoc) Ever since I was young, I wanted to hit the lottery and spend the rest of my life on private island (very realistic I know). Winning the lottery is a common dream among many Americans. The thought that anyone can win is enough to drive some people to buy the tickets. However, is it really just luck of the draw? Can any lucky person just win? Data science is modeling and visualizing data using plots and statistical analysis and I was able to obtain lottery data from New York state that can be visualized using techniques in this course.

Research questions:

1. Are certain numbers more likely to win the lottery?

- Is lucky number 7 significant in winning?
- Is my lucky number 3 significant in winning?
- If this is true, do they vary by game?

2. Is there a correlation between prize amount and numbers?

- Does the number 11 mean a larger payout?
- Does the number 6 mean a lower pay out?

3. What are the odds of winning each game normally?
4. How do number correlations vary from game to game?

- What are the number ranges for each number pulled? (The range for the first, the second, etc. . .)

5. What is the average prize?
6. Is there a correlation in the total of the winning numbers?
7. How significant is my data? Can I win the New York state lottery?

The plan of action may change based on how the data is presented from each source. The first step is going to be to look over the data and identify any possible issues that may arise. The next step will be to correct import and format the data to insure we are able to perform calculations. I will begin by calculating the number each number shows up in the winning numbers and creating a table of the values. I will calculate how often each number should show up if the values were truly random and compare this to the experimental values. I will then look at the correlation between the winning numbers. For data sets with additional

information, I will take into account that correlation in regard to that game(s). I will also take into account summary statistics. I will also heavily rely on calculations to determine If the findings are significant. Current data sources are shown below. Additional data sources may be added later. All data sources provide winning lottery numbers from New York over different periods.

Data:

1. Publisher State of New York. (2023a, July 28). *Lottery cash 4 life winning numbers: Beginning 2014*. Catalog. https://catalog.data.gov/dataset/lottery-cash-4-life-winning-numbers-beginning-2014
2. Publisher State of New York. (2023, July 28). *Lottery mega millions winning numbers: Beginning 2002*. Catalog. https://catalog.data.gov/dataset/lottery-mega-millions-winning-numbers-beginning-2002
3. Publisher State of New York. (2023b, July 28). *Lottery NY Lotto winning numbers: Beginning 2001*. Catalog. https://catalog.data.gov/dataset/lottery-ny-lotto-winning-numbers-beginning-2001
4. Publisher State of New York. (2023, July 28). *Lottery Powerball winning numbers: Beginning 2010*. Catalog. https://catalog.data.gov/dataset/lottery-powerball-winning-numbers-beginning-2010
5. Publisher State of New York. (2023e, July 28). *Lottery take 5 winning numbers*. Catalog. https://catalog.data.gov/dataset/lottery-take-5-winning-numbers

Likely require packages include ggplot2, a package to read the data depending on the format, pandoc, pcor, dplr, car, and more. Scatter plots, Q-Q plots, and histograms will be used to get a visual picture of the data and to analyze skewness and correlation. I am unsure at this moment of additional plots and packages that will help me in answering my question. I will have to research and become better at identifying ideal plots and packages. I will also have to research additional tests that could help me identify correlation between the lottery numbers. I also will have to consider additional visualization possibilities that I may not be comfortable with or that I have possible not yet heard of in this course.

Clean Data

```r
# Load required library
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# Load the data
setwd('/Users/gracieinman/Downloads')
Powerball_raw <- read.csv('Powerball.csv')
MegaMill_raw <- read.csv('MegaMill.csv')
Cash4Life_raw <- read.csv('Cash4Life.csv')
Take5_raw <- read.csv('Take5.csv')
NYlotto_raw <- read.csv('NYlotto.csv')

# Display the structure of the data
str(Powerball_raw)
```

```
## 'data.frame':    1499 obs. of  3 variables:
##  $ Draw.Date      : chr  "09/26/2020" "09/30/2020" "10/03/2020" "10/07/2020" ...
##  $ Winning.Numbers: chr  "11 21 27 36 62 24" "14 18 36 49 67 18" "18 31 36 43 47 20" "06 24 30 53 56
##  $ Multiplier     : int  3 2 2 2 3 2 2 3 2 2 ...
```

```
str(MegaMill_raw)
```

```
## 'data.frame':    2208 obs. of  4 variables:
##  $ Draw.Date      : chr  "09/25/2020" "09/29/2020" "10/02/2020" "10/06/2020" ...
##  $ Winning.Numbers: chr  "20 36 37 48 67" "14 39 43 44 67" "09 38 47 49 68" "15 16 18 39 59" ...
##  $ Mega.Ball      : int  16 19 25 17 13 25 12 10 22 1 ...
##  $ Multiplier     : int  2 3 2 3 2 2 5 3 2 3 ...
```

```
str(Cash4Life_raw)
```

```
## 'data.frame':    2007 obs. of  3 variables:
##  $ Draw.Date      : chr  "09/24/2020" "09/25/2020" "09/26/2020" "09/27/2020" ...
##  $ Winning.Numbers: chr  "18 20 43 45 60" "12 20 34 35 56" "12 15 27 45 46" "04 06 09 50 59" ...
##  $ Cash.Ball      : int  2 1 1 2 4 4 4 3 3 1 ...
```

```
str(Take5_raw)
```

```
## 'data.frame':    9588 obs. of  5 variables:
##  $ Draw.Date              : chr  "09/24/2020" "09/25/2020" "09/26/2020" "09/27/2020" ...
##  $ Evening.Winning.Numbers: chr  "02 05 10 15 18" "06 26 28 34 38" "11 12 13 19 36" "18 25 26 33 34"
##  $ Evening.Bonus..        : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Midday.Winning.Numbers : chr  "" "" "" "" ...
##  $ Midday.Bonus..         : logi  NA NA NA NA NA NA ...
```

```
str(NYlotto_raw)
```

```
## 'data.frame':    2281 obs. of  4 variables:
##  $ Draw.Date      : chr  "09/26/2020" "09/30/2020" "10/03/2020" "10/07/2020" ...
##  $ Winning.Numbers: chr  "14 19 20 23 55 57" "08 13 25 38 41 48" "19 20 24 42 44 56" "11 13 19 37 50
##  $ Bonus..        : int  22 42 37 8 6 48 30 38 34 59 ...
##  $ Extra..        : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
col_names <- colnames(Take5_raw)
print(col_names)
```

```
## [1] "Draw.Date"               "Evening.Winning.Numbers"
## [3] "Evening.Bonus.."         "Midday.Winning.Numbers"
## [5] "Midday.Bonus.."
```

```
# Handle missing values
Powerball <- na.omit(Powerball_raw)
MegaMill <- na.omit(MegaMill_raw)
Cash4Life <- na.omit(Cash4Life_raw)
NYLotto <- na.omit(NYlotto_raw)
# Remove empty columns
```

```r
columns_to_remove <- c("Midday.Winning.Numbers","Midday.Bonus..",
  "Evening.Bonus..")
Take5 <- Take5_raw[, !names(Take5_raw) %in% columns_to_remove]


#Show data
head(Powerball)
```

```
##    Draw.Date   Winning.Numbers Multiplier
## 1 09/26/2020 11 21 27 36 62 24          3
## 2 09/30/2020 14 18 36 49 67 18          2
## 3 10/03/2020 18 31 36 43 47 20          2
## 4 10/07/2020 06 24 30 53 56 19          2
## 5 10/10/2020 05 18 23 40 50 18          3
## 6 10/14/2020 21 37 52 53 58 05          2
```

```r
head(MegaMill)
```

```
##    Draw.Date Winning.Numbers Mega.Ball Multiplier
## 1 09/25/2020  20 36 37 48 67        16          2
## 2 09/29/2020  14 39 43 44 67        19          3
## 3 10/02/2020  09 38 47 49 68        25          2
## 4 10/06/2020  15 16 18 39 59        17          3
## 5 10/09/2020  05 11 25 27 64        13          2
## 6 10/13/2020  11 44 45 46 70        25          2
```

```r
head(Cash4Life)
```

```
##    Draw.Date Winning.Numbers Cash.Ball
## 1 09/24/2020  18 20 43 45 60         2
## 2 09/25/2020  12 20 34 35 56         1
## 3 09/26/2020  12 15 27 45 46         1
## 4 09/27/2020  04 06 09 50 59         2
## 5 09/28/2020  04 14 16 31 47         4
## 6 09/29/2020  06 10 37 46 52         4
```

```r
head(Take5)
```

```
##    Draw.Date Evening.Winning.Numbers
## 1 09/24/2020          02 05 10 15 18
## 2 09/25/2020          06 26 28 34 38
## 3 09/26/2020          11 12 13 19 36
## 4 09/27/2020          18 25 26 33 34
## 5 09/28/2020          04 24 31 32 33
## 6 09/29/2020          17 25 29 32 33
```

```r
head(NYLotto)
```

```
##       Draw.Date   Winning.Numbers Bonus.. Extra..
## 1138 05/29/2010 04 18 23 29 30 36      19      24
```

```
## 1139 05/26/2010 02 08 14 33 34 57       47       23
## 1140 05/22/2010 01 14 19 22 52 58       16       57
## 1141 05/19/2010 02 04 09 13 26 45       14       30
## 1142 05/15/2010 08 20 28 52 58 59       27       17
## 1143 05/12/2010 04 09 10 23 28 49       21        7
```

To clean the data set, I had to learn alternate methods to remove columns.
The Take5 dataset had three empty columns and previously known methods
were inaffective at removing the empty columns in the data. Instead they removed all the data in the datasets. I cleaned the data by omitting the NAs in most of the datasets. In the Take5 dataset, I had to remove empty columns. The data otherwise was clean for intial testing.

I plan on running multiple tests that may not seem like they apply to look at
to uncover possible correlations that I have not thought about. Plotting the data in different ways can help show patterns that I might not have seen by just looking at the data. I plan on looking up alternative packages and functions that I could possibly use to look at data that I had not thought of.

I am going to look at the winning numbers both individually and together. I plan on adding all the numbers to one data frame and comparing them that way.
I plan on looking at multipliers, bonus balls, and extra numbers individually due to the fact that not all of the data sets have them. Summarizing the full
dataset of winning numbers and summarizing them seperately to see how the data changes when adding them all together.I am going to look at the frequecy of numbers by themselves and in their positions. I am going to look at a variety off plots including scatter and histograms.I could also reduce by date by not looking at the multipliers or bonus numbers.

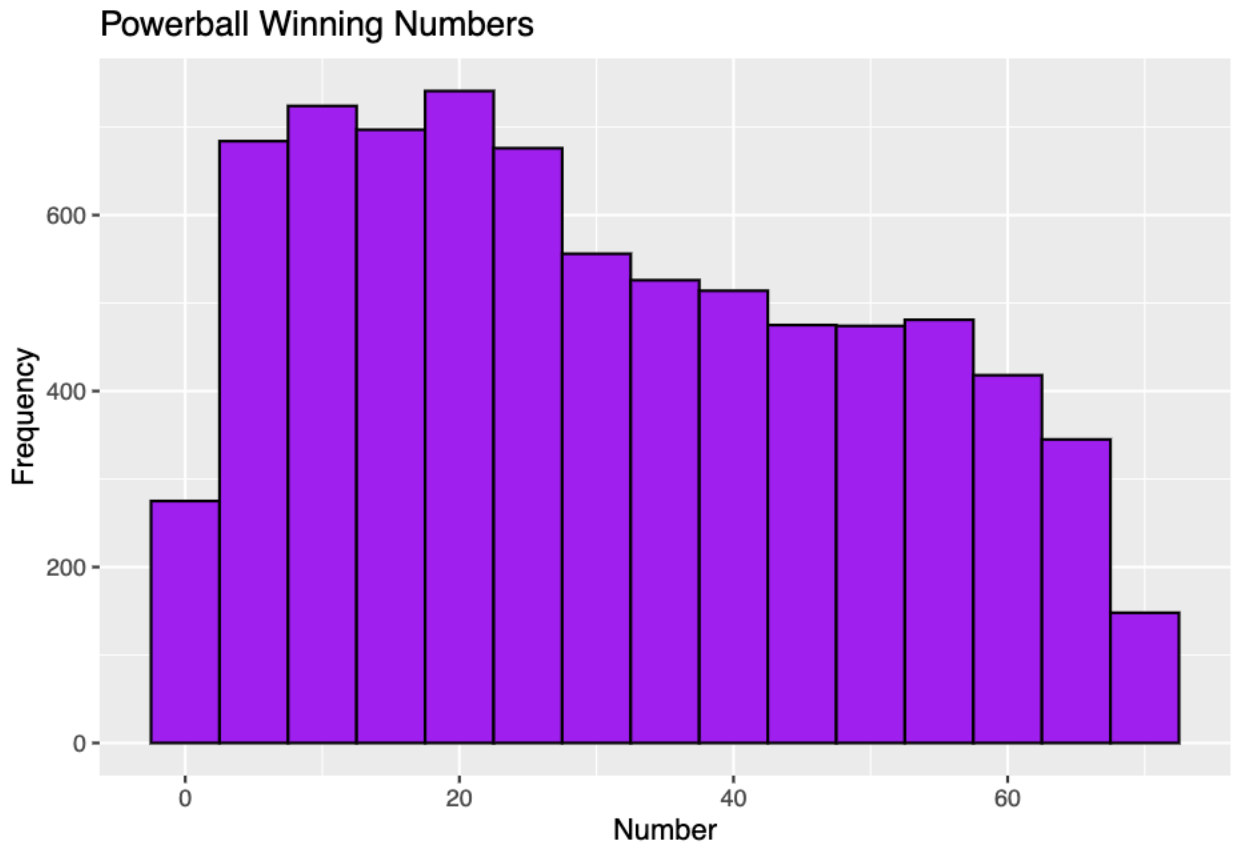
Grouping different categorical levels could allow me to analyze data in different ways by looking at the data differently in groups. Using the pipe opperator can make code more efficent by allowing you to perform multiple operations at once. This can help by improving the readablity of the code. I plan on looking more into machine learning tequniques. As of now I plan on incorporating regression into the analysis. I could filter out winning numbers to see how omiting them affects the results.

Separate the Winning.Numbers column into individual numbers

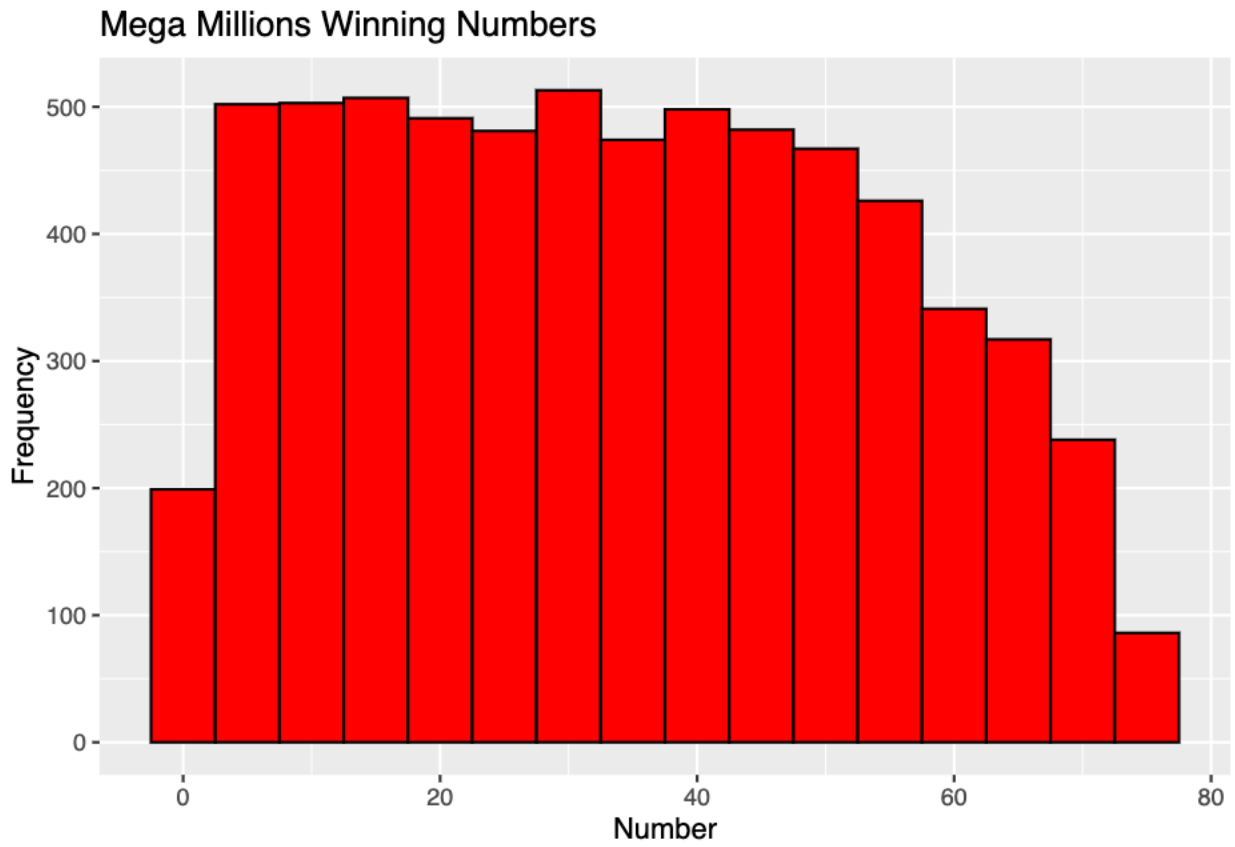
```
library(tidyr)
library(dplyr)
Powerball_win <- Powerball %>%
  separate_rows(Winning.Numbers, sep = " ") %>%
  mutate(Number = as.numeric(Winning.Numbers)) %>%
  select(Number)
MegaMill_win <- MegaMill %>%
  separate_rows(Winning.Numbers, sep = " ") %>%
  mutate(Number = as.numeric(Winning.Numbers)) %>%
  select(Number)
Cash4Life_win <- Cash4Life %>%
  separate_rows(Winning.Numbers, sep = " ") %>%
  mutate(Number = as.numeric(Winning.Numbers)) %>%
  select(Number)
Take5_win <- Take5 %>%
  separate_rows(Evening.Winning.Numbers, sep = " ") %>%
  mutate(Number = as.numeric(Evening.Winning.Numbers)) %>%
  select(Number)
NYLotto_win <- NYLotto %>%
```

```
  separate_rows(Winning.Numbers, sep = " ") %>%
  mutate(Number = as.numeric(Winning.Numbers)) %>%
  select(Number)
```

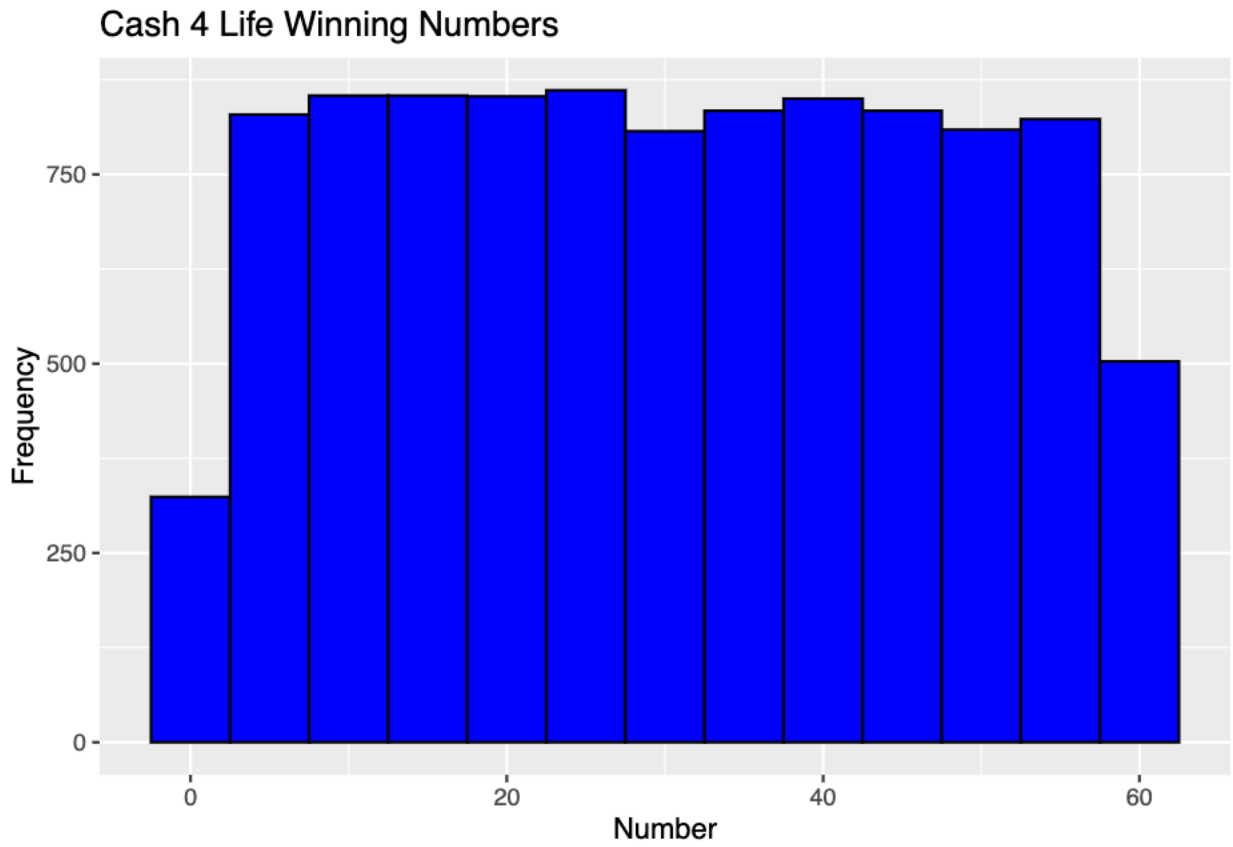
```
library(ggplot2)
Powerball_hist <- ggplot(Powerball_win, aes(x = Number)) +
  geom_histogram(binwidth = 5, fill = "purple", color = "black") +
  labs(title = "Powerball Winning Numbers", x = "Number", y = "Frequency")
print(Powerball_hist)
```



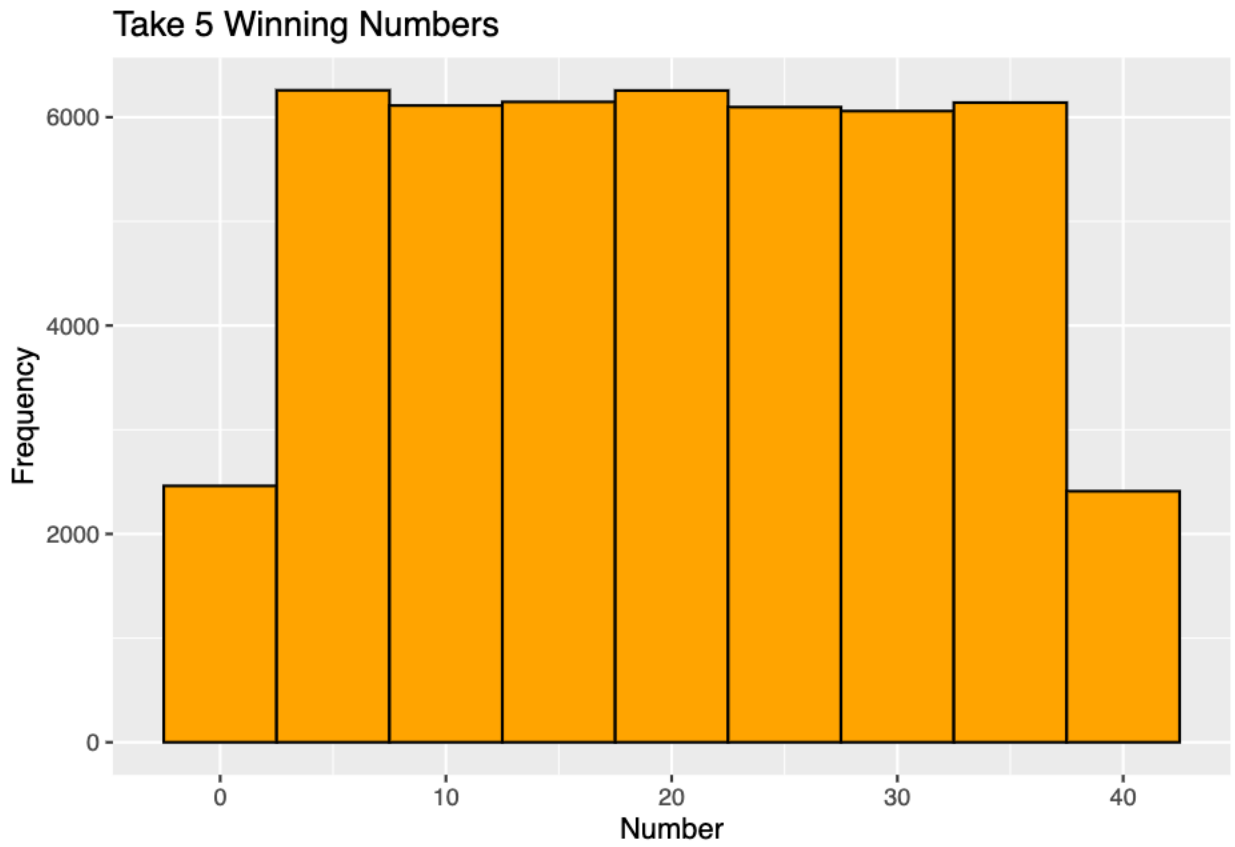

```
MegaMill_hist <- ggplot(MegaMill_win, aes(x = Number)) +
  geom_histogram(binwidth = 5, fill = "red", color = "black") +
  labs(title = "Mega Millions Winning Numbers", x = "Number", y = "Frequency")
print(MegaMill_hist)
```



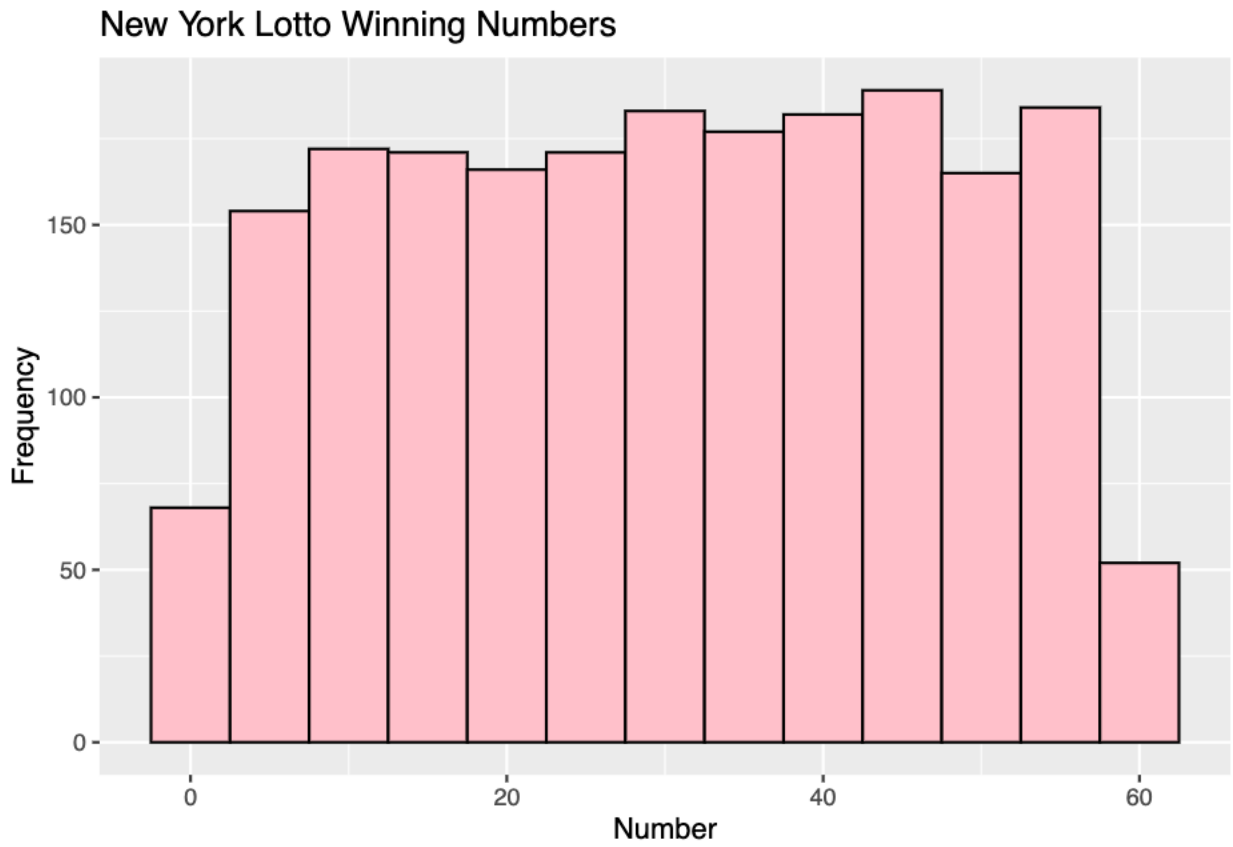

```
Cash4Life_hist <- ggplot(Cash4Life_win, aes(x = Number)) +
  geom_histogram(binwidth = 5, fill = "blue", color = "black") +
  labs(title = "Cash 4 Life Winning Numbers", x = "Number", y = "Frequency")
print(Cash4Life_hist)
```




```
Take5_hist <- ggplot(Take5_win, aes(x = Number)) +
  geom_histogram(binwidth = 5, fill = "orange", color = "black") +
  labs(title = "Take 5 Winning Numbers", x = "Number", y = "Frequency")
print(Take5_hist)
```




```
NYLotto_hist <- ggplot(NYLotto_win, aes(x = Number)) +
  geom_histogram(binwidth = 5, fill = "pink", color = "black") +
  labs(title = "New York Lotto Winning Numbers", x = "Number", y = "Frequency")
print(NYLotto_hist)
```




```
# Summary of datasets
cat("\n--- Powerball Summary ---\n")
```

```
##
## --- Powerball Summary ---
```

```
summary(Powerball_win)
```

```
##      Number
##  Min.   : 1.00
##  1st Qu.:14.00
##  Median :28.00
##  Mean   :30.76
##  3rd Qu.:47.00
##  Max.   :69.00
```

```
cat("\n--- Mega Millions Summary ---\n")
```

```
##
## --- Mega Millions Summary ---
```

```
summary(MegaMill_win)
```

```
##      Number
##  Min.   : 1.00
##  1st Qu.:17.00
##  Median :33.00
##  Mean   :33.93
##  3rd Qu.:50.00
##  Max.   :75.00
```

```
cat("\n--- Cash 4 Life Summary ---\n")
```

```
##
## --- Cash 4 Life Summary ---
```

```
summary(Cash4Life_win)
```

```
##      Number
##  Min.   : 1.00
##  1st Qu.:15.00
##  Median :30.00
##  Mean   :30.39
##  3rd Qu.:45.00
##  Max.   :60.00
```

```
cat("\n--- Take 5 Summary ---\n")
```

```
##
## --- Take 5 Summary ---
```

```
summary(Take5_win)
```

```
##      Number
##  Min.   : 1.00
##  1st Qu.:10.00
##  Median :20.00
##  Mean   :19.94
##  3rd Qu.:30.00
##  Max.   :39.00
```

```
cat("\n--- NY Lotto Summary ---\n")
```

```
##
## --- NY Lotto Summary ---
```

```
summary(NYLotto_win)
```

```
##      Number
##  Min.   : 1.00
##  1st Qu.:16.00
##  Median :31.00
##  Mean   :30.31
##  3rd Qu.:45.00
##  Max.   :59.00
```

```r
Powerball_sd <- sd(Powerball_win$Number)
cat("\nPowerball Standard Deviation: ", format(Powerball_sd, digits = 4), "\n")
```

```
##
## Powerball Standard Deviation:  19.27
```

```r
MegaMill_sd <- sd(MegaMill_win$Number)
cat("Mega Millions Standard Deviation: ", format(MegaMill_sd, digits = 4), "\n")
```

```
## Mega Millions Standard Deviation:  20.01
```

```r
Cash4Life_sd <- sd(Cash4Life_win$Number)
cat("Cash 4 Life Standard Deviation: ", format(Cash4Life_sd, digits = 4), "\n")
```

```
## Cash 4 Life Standard Deviation:  17.28
```

```r
Take5_sd <- sd(Take5_win$Number)
cat("Take 5 Standard Deviation: ", format(Take5_sd, digits = 4), "\n")
```

```
## Take 5 Standard Deviation:  11.24
```

```r
NYLotto_sd <- sd(NYLotto_win$Number)
cat("New York Lotto Standard Deviation: ", format(NYLotto_sd, digits = 4), "\n")
```

```
## New York Lotto Standard Deviation:  16.77
```

```r
# Function to test significance of each number in a dataset
test_number_significance <- function(data, alpha = 0.05) {
  results <- lapply(data, function(column) {
    t_test_result <- t.test(column, mu = mean(column))
    significance <- ifelse(t_test_result$p.value < alpha, "Significant", "Not Significant")
    result <- list(
      column_name = names(column),
      significance = significance,
      sample_mean = mean(column),
      p_value = t_test_result$p.value
    )
    return(result)
  })
  return(results)
}

# Calculate significance for each dataset
Powerball_significance <- test_number_significance(Powerball_win)
MegaMill_significance <- test_number_significance(MegaMill_win)
Cash4Life_significance <- test_number_significance(Cash4Life_win)
Take5_significance <- test_number_significance(Take5_win)
NYLotto_significance <- test_number_significance(NYLotto_win)
```

```r
print_results <- function(results, dataset_name) {
  cat("Significance Results for", dataset_name, "\n")
  cat("-------------------------\n")

  for (result in results) {
    cat("Number:", result$column_name, "\n")
    cat("Sample Mean:", result$sample_mean, "\n")
    cat("p-value:", result$p_value, "\n")
    cat("Significance:", result$significance, "\n")
    cat("\n")
  }
}
print_results(Powerball_significance, "Powerball")
```

```
## Significance Results for Powerball
## ------------------------
## Number:
## Sample Mean: 30.76429
## p-value: 1
## Significance: Not Significant
```

```r
print_results(MegaMill_significance, "Mega Millions")
```

```
## Significance Results for Mega Millions
## ------------------------
## Number:
## Sample Mean: 33.93057
## p-value: 1
## Significance: Not Significant
```

```r
print_results(Cash4Life_significance, "Cash 4 Life")
```

```
## Significance Results for Cash 4 Life
## ------------------------
## Number:
## Sample Mean: 30.38525
## p-value: 1
## Significance: Not Significant
```

```r
print_results(Take5_significance, "Take 5")
```

```
## Significance Results for Take 5
## ------------------------
## Number:
## Sample Mean: 19.93596
## p-value: 1
## Significance: Not Significant
```

```r
print_results(NYLotto_significance, "New York Lotto")
```

```
## Significance Results for New York Lotto
## ------------------------
## Number:
## Sample Mean: 30.31367
## p-value: 1
## Significance: Not Significant
```

Regression analysis

```
perform_regression_analysis <- function(data) {
  results <- lapply(data, function(column) {
    model <- lm(column ~ 1)
    result <- list(
      column_name = names(column),
      coefficients = coef(model),
      summary = summary(model)
    )
    return(result)
  })
  return(results)
}

Powerball_regression <- perform_regression_analysis(Powerball_win)
MegaMill_regression <- perform_regression_analysis(MegaMill_win)
Cash4Life_regression <- perform_regression_analysis(Cash4Life_win)
Take5_regression <- perform_regression_analysis(Take5_win)
NYLotto_regression <- perform_regression_analysis(NYLotto_win)

# Print the results
print_results <- function(results, dataset_name) {
  cat("Regression Analysis for", dataset_name, "\n")
  cat("------------------------\n")

  for (result in results) {
    cat("Variable:", result$column_name, "\n")
    cat("Coefficients:\n")
    print(result$coefficients)
    cat("Summary:\n")
    print(result$summary)
    cat("\n")
  }
}
print_results(Powerball_regression, "Powerball")
```

```
## Regression Analysis for Powerball
## ------------------------
## Variable:
## Coefficients:
## (Intercept)
##    30.76429
## Summary:
```

```
##
## Call:
## lm(formula = column ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.764 -16.764  -2.764  16.236  38.236
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.7643     0.2192   140.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.27 on 7733 degrees of freedom
```

```
print_results(MegaMill_regression, "Mega Millions")
```

```
## Regression Analysis for Mega Millions
## ------------------------
## Variable:
## Coefficients:
## (Intercept)
##    33.93057
## Summary:
##
## Call:
## lm(formula = column ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.931 -16.931  -0.931  16.069  41.069
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  33.9306     0.2477     137   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.01 on 6524 degrees of freedom
```

```
print_results(Cash4Life_regression, "Cash 4 Life")
```

```
## Regression Analysis for Cash 4 Life
## ------------------------
## Variable:
## Coefficients:
## (Intercept)
##    30.38525
## Summary:
##
## Call:
## lm(formula = column ~ 1)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.3853 -15.3853  -0.3853  14.6147  29.6147
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.3853     0.1725   176.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.28 on 10034 degrees of freedom
```

```
print_results(Take5_regression, "Take 5")
```

```
## Regression Analysis for Take 5
## ------------------------
## Variable:
## Coefficients:
## (Intercept)
##    19.93596
## Summary:
##
## Call:
## lm(formula = column ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.936  -9.936   0.064  10.064  19.064
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.93596    0.05136   388.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.24 on 47939 degrees of freedom
```

```
print_results(NYLotto_regression, "New York Lotto")
```

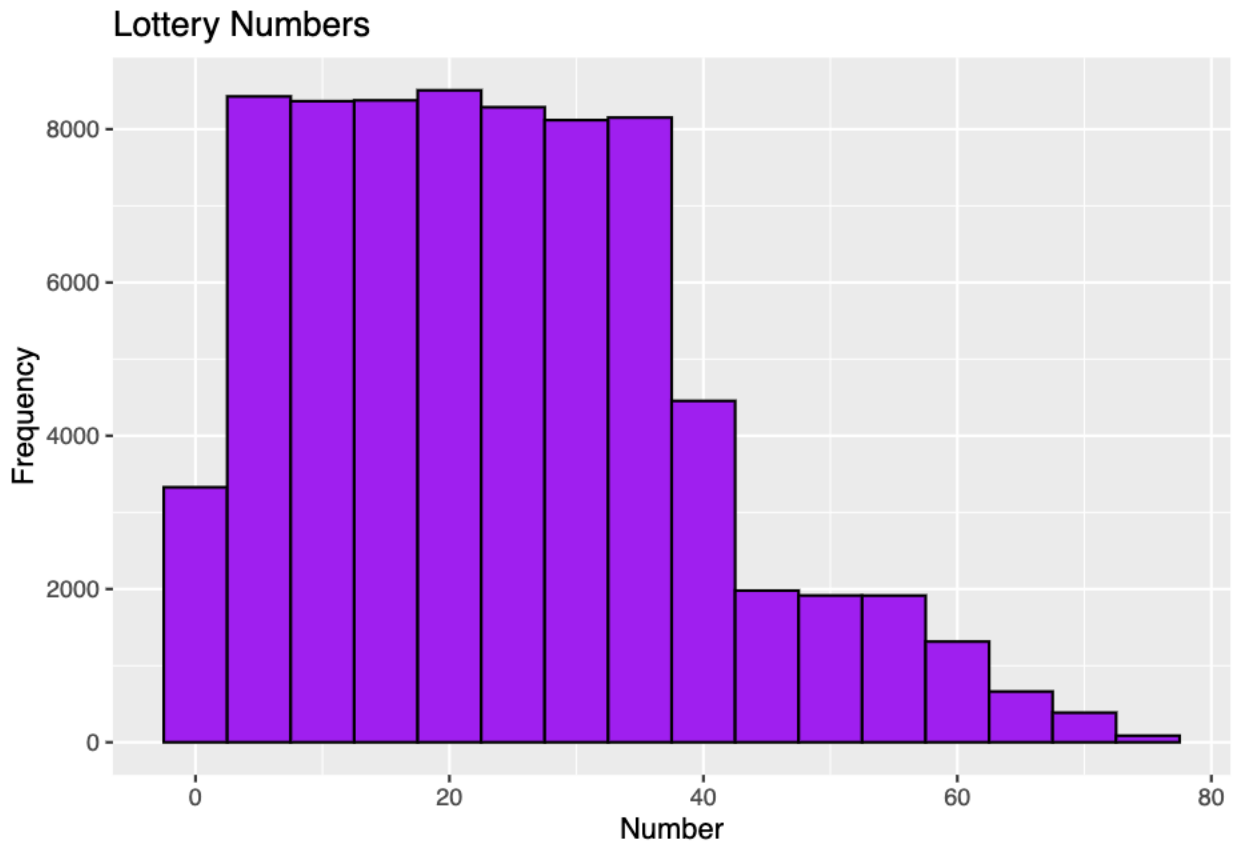
```
## Regression Analysis for New York Lotto
## ------------------------
## Variable:
## Coefficients:
## (Intercept)
##    30.31367
## Summary:
##
## Call:
## lm(formula = column ~ 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -29.3137 -14.3137   0.6863  14.6863  28.6863
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.3137     0.3719   81.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.77 on 2033 degrees of freedom
```

```r
all_win <- list(
  Powerball = Powerball_win$Number,
  MegaMill = MegaMill_win$Number,
  Cash4Life = Cash4Life_win$Number,
  Take5 = Take5_win$Number,
  NYLotto = NYLotto_win$Number
)

# Create a histogram of all the numbers
win_data <- data.frame(value = unlist(all_win))
Win_plot <- ggplot(win_data, aes(x = value)) +
  geom_histogram(binwidth = 5, fill = "purple", color = "black") +
  labs(title = "Lottery Numbers",
       x = "Number",
       y = "Frequency")

print(Win_plot)
```




```r
# Combine all winning numbers into a single vector
all_winning_numbers <- unlist(all_win)

# Count the frequency of each number
number_frequency <- table(all_winning_numbers)

# Sort the numbers by frequency in descending order
sorted_numbers <- sort(number_frequency, decreasing = TRUE)

# Print the 10 most common numbers
cat("The 10 Most Common Numbers:\n")
```

```
## The 10 Most Common Numbers:
```

```r
cat("-------------------------\n")
```

```
## -------------------------
```

```r
for (i in 1:10) {
  cat("Number:", names(sorted_numbers)[i], "\n")
  cat("Frequency:", sorted_numbers[i], "\n\n")
}
```

```
## Number: 6
```

```
## Frequency: 1750
##
## Number: 21
## Frequency: 1719
##
## Number: 17
## Frequency: 1711
##
## Number: 15
## Frequency: 1709
##
## Number: 31
## Frequency: 1709
##
## Number: 20
## Frequency: 1704
##
## Number: 22
## Frequency: 1700
##
## Number: 10
## Frequency: 1698
##
## Number: 7
## Frequency: 1697
##
## Number: 19
## Frequency: 1697
```

```r
# Perform regression analysis for each number in the dataset
perform_regression_analysis <- function(data) {
  results <- lapply(data, function(column) {
    model <- lm(column ~ 1)

    result <- list(
      column_name = names(column),
      coefficients = coef(model),
      summary = summary(model)
    )

    return(result)
  })

  return(results)
}

# Perform significance test for each number in the dataset
perform_significance_test <- function(data, alpha = 0.05) {
  results <- lapply(data, function(column) {
    t_test_result <- t.test(column, mu = mean(column))

    significance <- ifelse(t_test_result$p.value < alpha, "Significant", "Not Significant")

    result <- list(
```

```
      column_name = names(column),
      significance = significance,
      sample_mean = mean(column),
      p_value = t_test_result$p.value
    )

    return(result)
  })

  return(results)
}

# Perform regression analysis for win_data
win_data_regression <- perform_regression_analysis(win_data)

# Perform significance test for win_data
win_data_significance <- perform_significance_test(win_data)

# Print regression analysis results
print_results_regression <- function(results) {
  cat("Regression Analysis Results\n")
  cat("---------------------------\n")

  for (result in results) {
    cat("Variable:", result$column_name, "\n")
    cat("Coefficients:\n")
    print(result$coefficients)
    cat("Summary:\n")
    print(result$summary)
    cat("\n")
  }
}

# Print significance test results
print_results_significance <- function(results) {
  cat("Significance Test Results\n")
  cat("-------------------------\n")

  for (result in results) {
    cat("Number:", result$column_name, "\n")
    cat("Sample Mean:", result$sample_mean, "\n")
    cat("p-value:", result$p_value, "\n")
    cat("Significance:", result$significance, "\n")
    cat("\n")
  }
}

print_results_regression(win_data_regression)
```

```
## Regression Analysis Results
## ---------------------------
## Variable:
## Coefficients:
```

```
## (Intercept)
##    23.98923
## Summary:
##
## Call:
## lm(formula = column ~ 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.989 -11.989  -0.989  10.011  51.011
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.98923    0.05616   427.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.3 on 74267 degrees of freedom
```

```
print_results_significance(win_data_significance)
```

```
## Significance Test Results
## -------------------------
## Number:
## Sample Mean: 23.98923
## p-value: 1
## Significance: Not Significant
```

```
# Chances of winning by randomly choosing
total_balls <- 69 # number of balls in Powerball game
chosen_balls <- 5
matching_balls <- 5
calculate_winning_probability <- function(total_balls, chosen_balls, matching_balls) {
  total_combinations <- factorial(total_balls) / (factorial(chosen_balls) * factorial(total_balls - cho
  matching_combinations <- factorial(matching_balls) * factorial(chosen_balls - matching_balls)
  probability <- matching_combinations / total_combinations
  return(probability)
}
winning_probability <- calculate_winning_probability(total_balls, chosen_balls, matching_balls)
cat("Probability of winning: ", winning_probability, "\n")
```

```
## Probability of winning:  1.067757e-05
```

```
combinations_count <- list()

for (numbers in Powerball$Winning.Numbers) {
  sorted_numbers <- sort(numbers)  # Sort the numbers to ensure no specific order
  max_comb_size <- min(5, length(sorted_numbers))  # Set the maximum combination size
  for (r in 2:max_comb_size) {  # Count combinations of 2 to the maximum combination size
    if (length(sorted_numbers) >= r) {  # Check if there are enough numbers to form a combination of si
      combos <- combn(sorted_numbers, r)
      for (i in 1:ncol(combos)) {
```

```
        combo <- combos[, i]
        combo_str <- paste(combo, collapse = " ")
        combinations_count[[combo_str]] <- ifelse(is.null(combinations_count[[combo_str]]), 1, combinat
      }
    }
  }
}

# Convert combinations_count to a data frame
combinations_df <- data.frame(Combo = names(combinations_count), Count = unlist(combinations_count))

# Sort the data frame by Count in descending order
sorted_combinations <- combinations_df[order(combinations_df$Count, decreasing = TRUE), ]

# Print the top ten most frequent combinations
for (i in 1:min(10, nrow(sorted_combinations))) {
  combo_str <- sorted_combinations$Combo[i]
  count <- sorted_combinations$Count[i]
  cat(paste(combo_str, ": ", count, " times\n"))
}
```

```
## 11 21 27 36 62 24 :  1  times
## 14 18 36 49 67 18 :  1  times
## 18 31 36 43 47 20 :  1  times
## 06 24 30 53 56 19 :  1  times
## 05 18 23 40 50 18 :  1  times
## 21 37 52 53 58 05 :  1  times
## 06 10 31 37 44 23 :  1  times
## 01 03 13 44 56 26 :  1  times
## 18 20 27 45 65 06 :  1  times
## 11 28 37 40 53 13 :  1  times
```

```
combinations_count <- list()

for (numbers in MegaMill$Winning.Numbers) {
  sorted_numbers <- sort(numbers)  # Sort the numbers to ensure no specific order
  max_comb_size <- min(5, length(sorted_numbers))  # Set the maximum combination size
  for (r in 2:max_comb_size) {  # Count combinations of 2 to the maximum combination size
    if (length(sorted_numbers) >= r) {  # Check if there are enough numbers to form a combination of si
      combos <- combn(sorted_numbers, r)
      for (i in 1:ncol(combos)) {
        combo <- combos[, i]
        combo_str <- paste(combo, collapse = " ")
        combinations_count[[combo_str]] <- ifelse(is.null(combinations_count[[combo_str]]), 1, combinat
      }
    }
  }
}

# Convert combinations_count to a data frame
combinations_df <- data.frame(Combo = names(combinations_count), Count = unlist(combinations_count))

# Sort the data frame by Count in descending order
```

```r
sorted_combinations <- combinations_df[order(combinations_df$Count, decreasing = TRUE), ]

# Print the top ten most frequent combinations
for (i in 1:min(10, nrow(sorted_combinations))) {
  combo_str <- sorted_combinations$Combo[i]
  count <- sorted_combinations$Count[i]
  cat(paste(combo_str, ": ", count, " times\n"))
}
```

```
## 20 36 37 48 67 :  1  times
## 14 39 43 44 67 :  1  times
## 09 38 47 49 68 :  1  times
## 15 16 18 39 59 :  1  times
## 05 11 25 27 64 :  1  times
## 11 44 45 46 70 :  1  times
## 27 32 50 52 57 :  1  times
## 46 54 57 58 66 :  1  times
## 18 34 44 60 69 :  1  times
## 06 13 34 46 62 :  1  times
```

```r
combinations_count <- list()

for (numbers in Cash4Life$Winning.Numbers) {
  sorted_numbers <- sort(numbers)  # Sort the numbers to ensure no specific order
  max_comb_size <- min(5, length(sorted_numbers))  # Set the maximum combination size
  for (r in 2:max_comb_size) {  # Count combinations of 2 to the maximum combination size
    if (length(sorted_numbers) >= r) {  # Check if there are enough numbers to form a combination of si
      combos <- combn(sorted_numbers, r)
      for (i in 1:ncol(combos)) {
        combo <- combos[, i]
        combo_str <- paste(combo, collapse = " ")
        combinations_count[[combo_str]] <- ifelse(is.null(combinations_count[[combo_str]]), 1, combinat
      }
    }
  }
}

# Convert combinations_count to a data frame
combinations_df <- data.frame(Combo = names(combinations_count), Count = unlist(combinations_count))

# Sort the data frame by Count in descending order
sorted_combinations <- combinations_df[order(combinations_df$Count, decreasing = TRUE), ]

# Print the top ten most frequent combinations
for (i in 1:min(10, nrow(sorted_combinations))) {
  combo_str <- sorted_combinations$Combo[i]
  count <- sorted_combinations$Count[i]
  cat(paste(combo_str, ": ", count, " times\n"))
}
```

```
## 18 20 43 45 60 :  1  times
## 12 20 34 35 56 :  1  times
## 12 15 27 45 46 :  1  times
```

```
## 04 06 09 50 59 :  1  times
## 04 14 16 31 47 :  1  times
## 06 10 37 46 52 :  1  times
## 01 16 39 41 58 :  1  times
## 07 23 37 49 52 :  1  times
## 02 26 29 35 53 :  1  times
## 05 08 42 45 47 :  1  times
```

```
combinations_count <- list()

for (numbers in Take5$Evening.Winning.Numbers) {
  sorted_numbers <- sort(numbers)  # Sort the numbers to ensure no specific order
  max_comb_size <- min(5, length(sorted_numbers))  # Set the maximum combination size
  for (r in 2:max_comb_size) {  # Count combinations of 2 to the maximum combination size
    if (length(sorted_numbers) >= r) {  # Check if there are enough numbers to form a combination of si
      combos <- combn(sorted_numbers, r)
      for (i in 1:ncol(combos)) {
        combo <- combos[, i]
        combo_str <- paste(combo, collapse = " ")
        combinations_count[[combo_str]] <- ifelse(is.null(combinations_count[[combo_str]]), 1, combinat
      }
    }
  }
}

# Convert combinations_count to a data frame
combinations_df <- data.frame(Combo = names(combinations_count), Count = unlist(combinations_count))

# Sort the data frame by Count in descending order
sorted_combinations <- combinations_df[order(combinations_df$Count, decreasing = TRUE), ]

# Print the top ten most frequent combinations
for (i in 1:min(10, nrow(sorted_combinations))) {
  combo_str <- sorted_combinations$Combo[i]
  count <- sorted_combinations$Count[i]
  cat(paste(combo_str, ": ", count, " times\n"))
}
```

```
## 02 19 23 24 37 :  2  times
## 06 07 10 16 27 :  2  times
## 05 12 17 26 30 :  2  times
## 01 13 15 18 29 :  2  times
## 07 11 15 16 28 :  2  times
## 18 21 27 37 38 :  2  times
## 07 09 20 26 31 :  2  times
## 01 18 19 31 32 :  2  times
## 01 04 06 17 31 :  2  times
## 01 10 22 27 35 :  2  times
```

```
combinations_count <- list()

for (numbers in NYLotto$Winning.Numbers) {
  sorted_numbers <- sort(numbers)  # Sort the numbers to ensure no specific order
```

```r
  max_comb_size <- min(5, length(sorted_numbers))
  for (r in 2:max_comb_size)
    if (length(sorted_numbers) >= r) {
      combos <- combn(sorted_numbers, r)
      for (i in 1:ncol(combos)) {
        combo <- combos[, i]
        combo_str <- paste(combo, collapse = " ")
        combinations_count[[combo_str]] <- ifelse(is.null(combinations_count[[combo_str]]), 1, combinat
      }
    }
  }

# Convert combinations_count to a data frame
combinations_df <- data.frame(Combo = names(combinations_count), Count = unlist(combinations_count))

# Sort the data frame by Count in descending order
sorted_combinations <- combinations_df[order(combinations_df$Count, decreasing = TRUE), ]

# Print the top ten most frequent combinations
for (i in 1:min(10, nrow(sorted_combinations))) {
  combo_str <- sorted_combinations$Combo[i]
  count <- sorted_combinations$Count[i]
  cat(paste(combo_str, ": ", count, " times\n"))
}
```

```
## 04 18 23 29 30 36 :  1  times
## 02 08 14 33 34 57 :  1  times
## 01 14 19 22 52 58 :  1  times
## 02 04 09 13 26 45 :  1  times
## 08 20 28 52 58 59 :  1  times
## 04 09 10 23 28 49 :  1  times
## 07 31 33 38 45 46 :  1  times
## 08 11 15 24 52 53 :  1  times
## 04 05 12 23 43 56 :  1  times
## 01 07 15 19 49 50 :  1  times
```