Wrangling datasets through a range of operations including column and row selection; data filtering, sorting, and augmenting; and summarizing the data utilization using built-in descriptive statistics.

Project Report-2 Submitted By

Iti Rohilla

Northeastern University Boston Campus

ALY6000 70917 Introduction to Analytics SEC 19 Fall 2023 CPS [BOS-A-HY]

By Instructor Roy Wada

Submitted On

Oct-03-2023

SUMMARY

Data can measure many things: gross domestic product (GDP), baseball statistics, and more. In any arena, data analysis is a numbers game played at different scales. Skilled data analysts work confidently with data from all content areas to surface both patterns and trends, as well as discrete data points. Practice locating patterns in two different sets of data. Specifically, to learn how R facilitates operations like selecting columns, filtering, and adding rows to quickly arrive at meaningful conclusions and answers.

Objectives:

cleaning and wrangling a dataset enable better data analysis.

Use grouping and faceting to create multivariate graphs.

Manipulate dates and missing values.

Understand data type conversions.

Encounter the ggplot2 package.

Sort, merge, and subset datasets

Create and recode variables.

Select and drop variables.

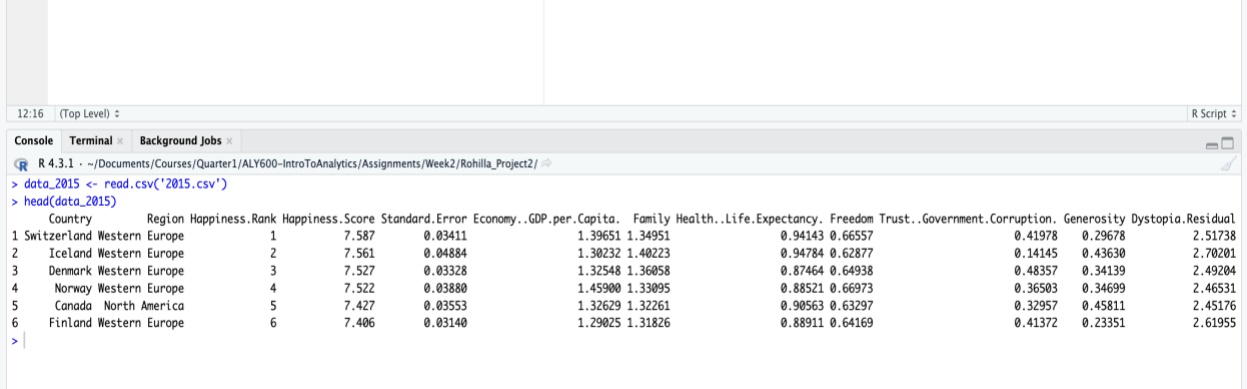
Create a simple bivariate (two-variable graph

Save graphs in multiple formats.

##########**Assignment-1**##########

Q1. Read the data set 2015.csv and store it in a variable called data_2015. You can test that you loaded it correctly with the code utilizing the head function below.

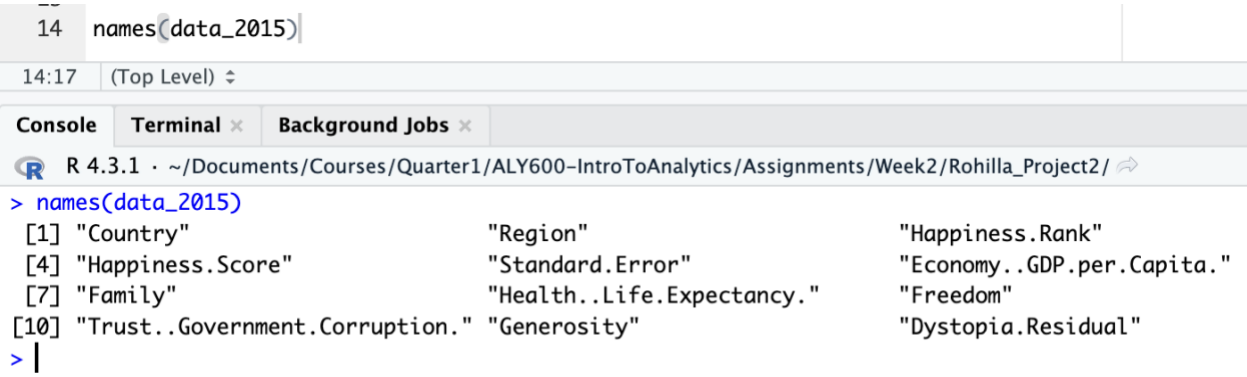
A1.



```
12:16  (Top Level)                                                    R Script
Console  Terminal  Background Jobs
R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/
> data_2015 <- read.csv('2015.csv')
> head(data_2015)
      Country        Region Happiness.Rank Happiness.Score Standard.Error Economy..GDP.per.Capita.  Family Health..Life.Expectancy. Freedom Trust..Government.Corruption. Generosity Dystopia.Residual
1 Switzerland Western Europe             1           7.587        0.03411                 1.39651 1.34951                 0.94143 0.66557                       0.41978    0.29678           2.51738
2     Iceland Western Europe             2           7.561        0.04884                 1.30232 1.40223                 0.94784 0.62877                       0.14145    0.43630           2.70201
3     Denmark Western Europe             3           7.527        0.03328                 1.32548 1.36058                 0.87464 0.64938                       0.48357    0.34139           2.49204
4      Norway Western Europe             4           7.522        0.03880                 1.45900 1.33095                 0.88521 0.66973                       0.36503    0.34699           2.46531
5      Canada  North America             5           7.427        0.03553                 1.32629 1.32261                 0.90563 0.63297                       0.32957    0.45811           2.45176
6     Finland Western Europe             6           7.406        0.03140                 1.29025 1.31826                 0.88911 0.64169                       0.41372    0.23351           2.61955
>
```

Q2. Use the function names to produce the column names for your data set.

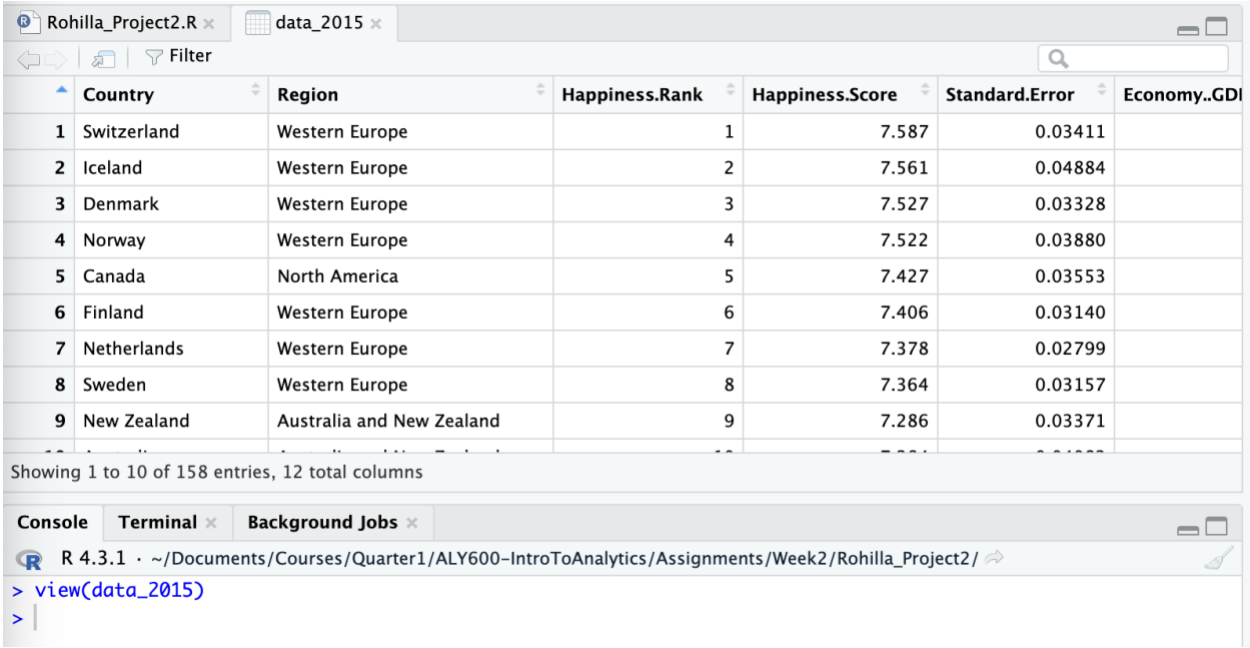
A2.



```
14  names(data_2015)
14:17  (Top Level)
Console  Terminal  Background Jobs
R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/
> names(data_2015)
 [1] "Country"                    "Region"                     "Happiness.Rank"
 [4] "Happiness.Score"            "Standard.Error"             "Economy..GDP.per.Capita."
 [7] "Family"                     "Health..Life.Expectancy."   "Freedom"
[10] "Trust..Government.Corruption." "Generosity"                "Dystopia.Residual"
>
```

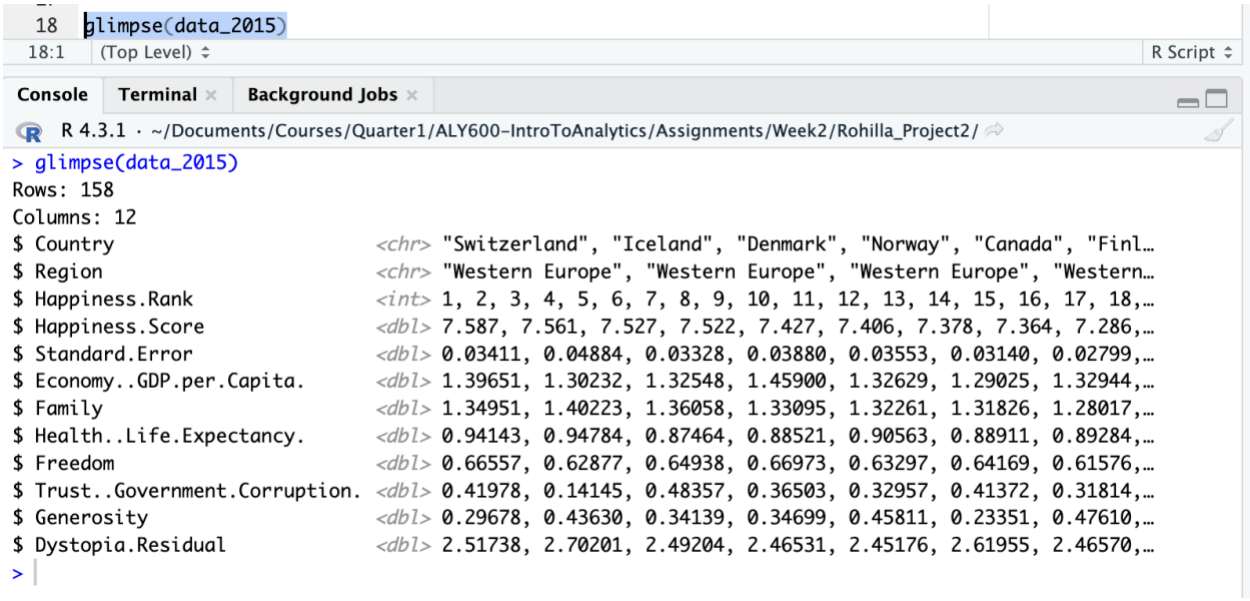
Q3. Use the view function to view the data set in a separate tab.

A3.



Q4. Use the glimpse function to view your data set in another configuration.

A4.



Q5. Use p_load to install the janitor package. Janitor has a function called clean_names that can be given a data frame to make the names more R friendly. Be sure to store the resulting converted data frame in a variable.

A5.

```
20  p_load(janitor)
21  data_2015 <- clean_names(data_2015)
22  data_2015
23
24
```

23:1 (Top Level)

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

50	0.02901	0.22823	2.02518
51	0.08800	0.20536	2.82334
52	0.01615	0.20951	3.10712
53	0.08242	0.34240	2.18896
54	0.08454	0.11827	2.24729
55	0.03787	0.25328	1.61583
56	0.01031	0.02641	2.44649
57	0.19317	0.27815	2.32407
58	0.05989	0.14982	2.59450
59	0.19090	0.11046	2.13090
60	0.04212	0.16759	1.86565
61	0.10501	0.33075	1.88541
62	0.02430	0.05444	2.75414
63	0.11023	0.18295	2.09066
64	0.03005	0.00199	2.27394
65	0.02299	0.21230	2.32038
66	0.14280	0.26169	1.59888
67	0.06146	0.30638	1.88931
68	0.17383	0.07822	2.43209
69	0.04741	0.28310	2.76579
70	0.30844	0.16979	1.86984
71	0.07521	0.37744	1.76145
72	0.37124	0.39478	0.65429
73	0.15184	0.08680	1.58782
74	0.00000	0.51535	1.86399
75	0.10441	0.16860	2.20173
76	0.15746	0.12253	2.08528
77	0.04232	0.30030	2.23270
78	0.04030	0.27233	2.89319
79	0.15445	0.47998	1.63794
80	0.16065	0.07799	2.00073
81	0.10464	0.33671	3.10709
82	0.14293	0.11053	1.87996
83	0.14296	0.16140	2.10017

Q6. Select from the data set the country, region, happiness_score, and freedom columns. Store this new table as happy_df.

A6.

```
24  happy_df <- data_2015 %>%
25    select(country, region, happiness_score, freedom)
26  happy_df
27
```

27:1 (Top Level)

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> happy_df <- data_2015 %>%
+   select(country, region, happiness_score, freedom)
> happy_df
                  country                       region happiness_score freedom
1             Switzerland               Western Europe           7.587 0.66557
2                 Iceland               Western Europe           7.561 0.62877
3                 Denmark               Western Europe           7.527 0.64938
4                  Norway               Western Europe           7.522 0.66973
5                  Canada                North America           7.427 0.63297
6                 Finland               Western Europe           7.406 0.64169
```

Q7. Slice the first 10 rows from happy_df and store it as top_ten_df.

A7.

```
28  top_ten_df <- happy_df %>%
29    slice(1:10)
30  top_ten_df
31
```

23:1 (Top Level)

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> top_ten_df <- happy_df %>%
+   slice(1:10)
> top_ten_df
       country                   region happiness_score freedom
1  Switzerland           Western Europe           7.587 0.66557
2      Iceland           Western Europe           7.561 0.62877
3      Denmark           Western Europe           7.527 0.64938
4       Norway           Western Europe           7.522 0.66973
5       Canada            North America           7.427 0.63297
6      Finland           Western Europe           7.406 0.64169
7  Netherlands           Western Europe           7.378 0.61576
8       Sweden           Western Europe           7.364 0.65980
9  New Zealand Australia and New Zealand           7.286 0.63938
10   Australia Australia and New Zealand           7.284 0.65124
>
```

Q8. From happy_df filter the table for freedom values under 0.20. Store this new table as no_freedom_df.

A8.

```
32  no_freedom_df <- happy_df %>%
33    filter(freedom<0.20)
34  no_freedom_df
```

```
34:15  (Top Level)
```

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> no_freedom_df <- happy_df %>%
+   filter(freedom<0.20)
> no_freedom_df
                country                          region happiness_score freedom
1              Pakistan                      Southern Asia           5.194 0.12102
2            Montenegro       Central and Eastern Europe           5.192 0.18260
3  Bosnia and Herzegovina       Central and Eastern Europe           4.949 0.09245
4                Greece                    Western Europe           4.857 0.07699
5                  Iraq Middle East and Northern Africa           4.677 0.00000
6                 Sudan               Sub-Saharan Africa           4.550 0.10081
7               Armenia       Central and Eastern Europe           4.350 0.19847
8                 Egypt Middle East and Northern Africa           4.194 0.17288
9                Angola               Sub-Saharan Africa           4.033 0.10384
10           Madagascar               Sub-Saharan Africa           3.681 0.19184
11                Syria Middle East and Northern Africa           3.006 0.15684
12              Burundi               Sub-Saharan Africa           2.905 0.11850
>
```

Q9. Arrange the values in happy_df in descending order by their freedom values. Store this new table as best_freedom_df.

A9.

```
42  best_freedom_df <- happy_df %>%
43    arrange(desc(freedom))
44  best_freedom_df
45
46
```

```
42:1  (Top Level)
```

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
128                    Latvia       Central and Eastern Europe           5.098 0.29671
129                   Algeria Middle East and Northern Africa           5.605 0.28579
130                   Liberia               Sub-Saharan Africa           4.571 0.28531
131                   Tunisia Middle East and Northern Africa           4.739 0.26268
132                     Italy                    Western Europe           5.948 0.26236
133                   Croatia       Central and Eastern Europe           5.759 0.25883
134                  Zimbabwe               Sub-Saharan Africa           4.610 0.25861
135                   Ukraine       Central and Eastern Europe           4.681 0.25123
136                    Kosovo       Central and Eastern Europe           5.589 0.24749
137  Palestinian Territories Middle East and Northern Africa           4.715 0.24499
138                     Haiti      Latin America and Caribbean           4.518 0.24425
139                Mauritania               Sub-Saharan Africa           4.436 0.24232
140                     Chad               Sub-Saharan Africa           3.667 0.23501
141               Afghanistan                     Southern Asia           3.575 0.23414
142                   Comoros               Sub-Saharan Africa           3.956 0.22917
143                    Turkey Middle East and Northern Africa           5.332 0.22815
```

Q10. Create a new column with mutate in data_2015 called gff_stat. For each row, the gff_stat is the sum of the family, freedom, and generosity values. Store the resulting table right in the data_2015 variable.

A10.

```
46  data_2015  <- data_2015 %>%
47    mutate(gff_stat =family+freedom+generosity)
48  data_2015
49
50
51
52
53
54
55
```

51:1 (Top Level)

Console | Terminal | Background Jobs

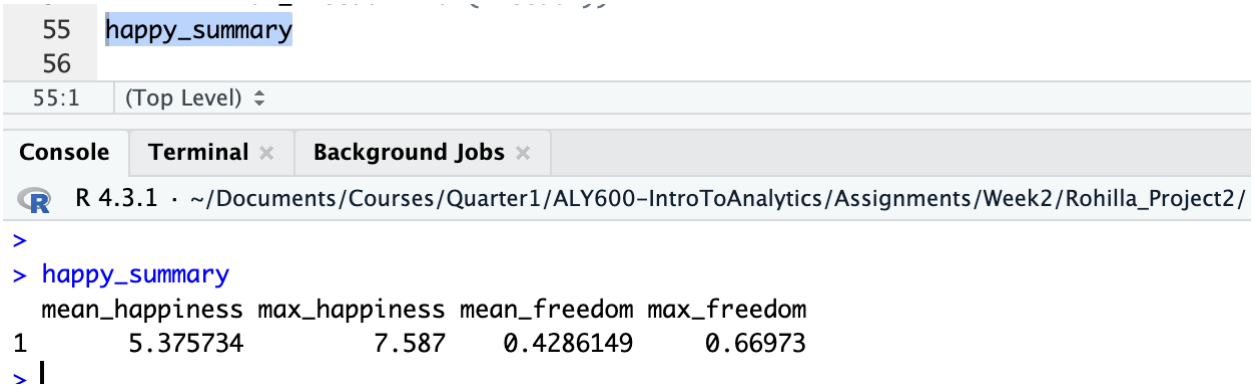
R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAn

	dystopia_residual	gff_stat
1	2.51738	2.31186
2	2.70201	2.46730
3	2.49204	2.35135
4	2.46531	2.34767
5	2.45176	2.41369
6	2.61955	2.19346
7	2.46570	2.37203
8	2.37119	2.31149
9	2.26425	2.43406
10	2.26646	2.39609
11	3.08854	1.96884

Q11. Summarize the happy_df data set. Your summary should contain the mean happiness_score in a column called mean_happiness, the max happiness_score in a column called max_happiness, the mean freedom in a column called

mean_freedom, and the max freedom in a column called max_freedom. Store the resulting table as happy_summary.

A11.



Q12. Group the happy_df data set by region. Run a summary that provides the number of countries in each region in a column called country_count, the mean happiness for each region in a column called mean_happiness, and the mean freedom of each region in a column called mean_freedom. Store your resulting table in a variable called regional_stats_df.

A12.

```
57 regional_stats_df <- happy_df %>%
58   group_by(region) %>%
59   summarise(
60     country_count = n(),
61     mean_happiness = mean(happiness_score),
62     mean_freedom = mean(freedom)
63      )
64 regional_stats_df
```

64:1 (Top Level)

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> regional_stats_df <- happy_df %>%
+   group_by(region) %>%
+   summarise(
+     country_count = n(),
+     mean_happiness = mean(happiness_score),
+     mean_freedom = mean(freedom)
+      )
> regional_stats_df
# A tibble: 10 × 4
   region                          country_count mean_happiness mean_freedom
   <chr>                                   <int>          <dbl>        <dbl>
 1 Australia and New Zealand                   2           7.28        0.645
 2 Central and Eastern Europe                 29           5.33        0.358
 3 Eastern Asia                                6           5.63        0.462
 4 Latin America and Caribbean                22           6.14        0.502
 5 Middle East and Northern Africa            20           5.41        0.362
 6 North America                               2           7.27        0.590
 7 Southeastern Asia                           9           5.32        0.557
 8 Southern Asia                               7           4.58        0.373
 9 Sub-Saharan Africa                         40           4.20        0.366
10 Western Europe                             21           6.69        0.550
>
```

Q13. [Challenge Problem] Compare the average gdp per capita of the ten least happy Western European countries with the ten happiest Sub-Saharan African countries.

For testing, you can store the resulting data.frame or table as gdp_df.

A13.

```
WE <- data_2015 %>%
  filter(region == 'Western Europe')  %>%
  arrange(happiness_score)  %>%
  head(10)
SSA <- data_2015 %>%
  filter(region == 'Sub-Saharan Africa')  %>%
  arrange(desc(happiness_score))  %>%
  head(10)

avg_gdp_we <- mean(WE$economy_gdp_per_capita)
avg_gdp_ssa <- mean(SSA$economy_gdp_per_capita)

gdp_df <-  data.frame(
  europe_gdp = c(avg_gdp_we),
  africa_gdp = c(avg_gdp_ssa)
)
gdp_df
```

82:7 (Top Level)

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> WE <- data_2015 %>%
+   filter(region == 'Western Europe')  %>%
+   arrange(happiness_score)  %>%
+   head(10)
> SSA <- data_2015 %>%
+   filter(region == 'Sub-Saharan Africa')  %>%
+   arrange(desc(happiness_score))  %>%
+   head(10)
>
> avg_gdp_we <- mean(WE$economy_gdp_per_capita)
> avg_gdp_ssa <- mean(SSA$economy_gdp_per_capita)
>
> gdp_df <-  data.frame(
+   europe_gdp = c(avg_gdp_we),
+   africa_gdp = c(avg_gdp_ssa)
+ )
> gdp_df
  europe_gdp africa_gdp
1   1.229088   0.522847
>
```

Q14. [Challenge problem] From your regional_stats_df, create a scatterplot of mean_happiness vs. mean_freedom. Draw a line segment from the smallest of these values to the largest.

A14.

```r
min_happiness <- min(regional_stats_df$mean_happiness)
max_happiness <- max(regional_stats_df$mean_happiness)
min_freedom <- min(regional_stats_df$mean_freedom)
max_freedom <- max(regional_stats_df$mean_freedom)

custom_colors <- c(
  "#FF0000", "#FF9933", "#666633", "#009933", "#33CC66",
  "#339999", "#3399FF", "#DA70D6", "#FF69B4", "#FF1493"
)

ggplot(data = regional_stats_df, aes(x = mean_happiness, y = mean_freedom, color = region)) +
  geom_point() +  # Add points for the scatterplot
  scale_color_manual(
    values = custom_colors,
    breaks = unique(regional_stats_df$region),
    labels = unique(regional_stats_df$region)
  ) +
  geom_segment(aes(x = min_happiness, y = min_freedom, xend = max_happiness, yend = max_freedom), color = "black")
```

104:1 (Top Level) R Script

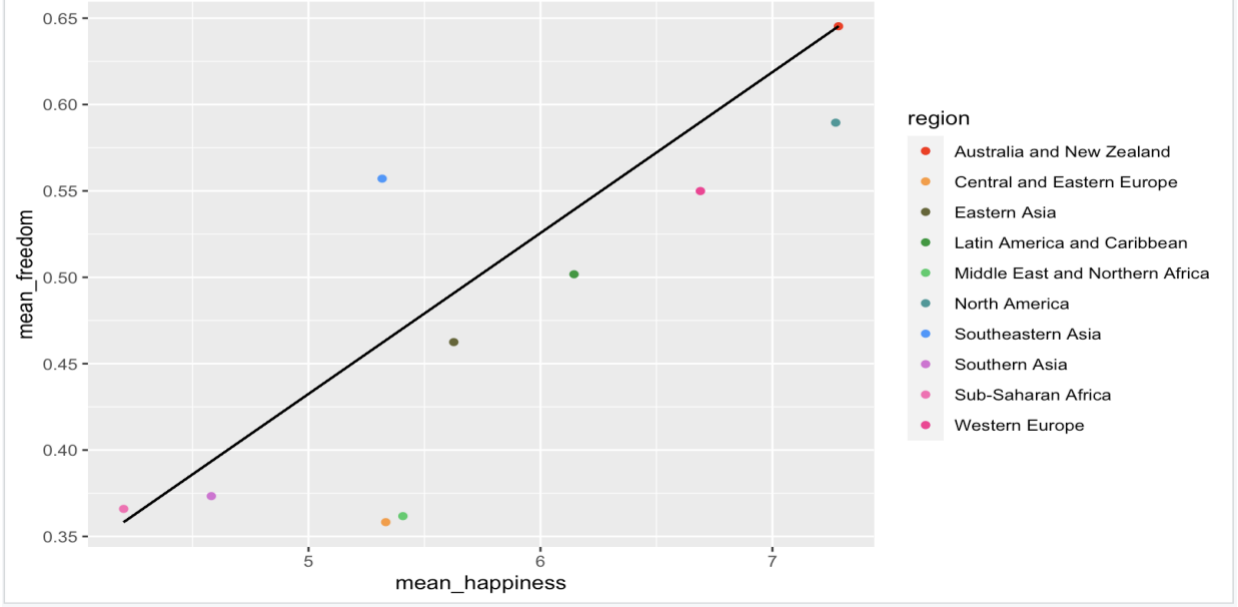
Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> min_happiness <- min(regional_stats_df$mean_happiness)
> max_happiness <- max(regional_stats_df$mean_happiness)
> min_freedom <- min(regional_stats_df$mean_freedom)
> max_freedom <- max(regional_stats_df$mean_freedom)
>
> custom_colors <- c(
+   "#FF0000", "#FF9933", "#666633", "#009933", "#33CC66",
+   "#339999", "#3399FF", "#DA70D6", "#FF69B4", "#FF1493"
+ )
>
> ggplot(data = regional_stats_df, aes(x = mean_happiness, y = mean_freedom, color = region)) +
+   geom_point() +  # Add points for the scatterplot
+   scale_color_manual(
+     values = custom_colors,
+     breaks = unique(regional_stats_df$region),
+     labels = unique(regional_stats_df$region)
+   ) +
+   geom_segment(aes(x = min_happiness, y = min_freedom, xend = max_happiness, yend = max_freedom), color = "black")
>
```

Files Plots Packages Help Viewer Presentation

Zoom Export Publish



##########**Assignment-2**##########

Q1. Download the baseball.csv data set. data set that represents batting statistics from the 1986 Major League Baseball season. Read this data set in a variable called baseball.

A1.

```
105 ##################################Assignmnet2##
106
107 baseball <- read.csv('baseball.csv')
108
107:1  # Assignmnet2
```

```
Console  Terminal  Background Jobs
R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/A
> baseball <- read.csv('baseball.csv')
>
```

Q3. Use the class function to discover the type of class represented in the baseball dataset.

A3.

```
109 class_types <- class(baseball)
110 class_types
111
112
113
114
110:12  # Assignmnet2
```

```
Console  Terminal  Background Jobs
R 4.3.1 · ~/Documents/Courses/Quarter1/
> class_types <- class(baseball)
> class_types
[1] "data.frame"
>
```

Q4. For each age, compute the following: the number of people at that age, the average number of home runs (HRs), the average number of hits, and the average number of runs scored. Store these computations in a variable called age_stats_df.

A4.

```
112  age_stats_df <- baseball %>%
113    group_by(Age) %>%
114    summarise(
115      Count = n(),
116      HR = mean(HR, na.rm = TRUE),
117      H = mean(H, na.rm = TRUE),
118      R = mean(R, na.rm = TRUE)
119    )
120  age_stats_df
```

120:1 # Assignmnet2

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-Ir

```
> age_stats_df
# A tibble: 24 × 5
     Age Count    HR     H     R
   <int> <int> <dbl> <dbl> <dbl>
 1    20     5  3.4   24    11.8
 2    21    18  3.28  22.4  14.1
 3    22    38  2.32  28.5  14.3
 4    23    38  3.74  36.7  20.0
 5    24    65  4.37  42.6  22.1
 6    25    94  4.5   42.8  21.0
 7    26    86  5.70  49.8  24.9
 8    27    63  4.62  52.0  27.1
 9    28    64  3.94  49.3  25.8
10    29    53  5.26  52.6  26.4
# ℹ 14 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

Q5. Remove (filter) from baseball any player with 0 at bats (AB). Store the result in baseball.

A5.

```
122
123 baseball <- baseball %>%
124   filter(AB > 0)
125 baseball
125:1  # Assignmnet2
```

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rol

```
> baseball
          Last  First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS  BB  SO
1        Acker    Jim  27  21  28  28   1   3   1   0  0   0  0  0   0  21
2       Adduci    Jim  26   3  13  11   2   1   1   0  0   0  0  0   1   2
3       Aguayo   Luis  27  62 146 133  17  28   6   1  4  13  1  1   8  26
4     Aguilera   Rick  24  32  57  51   4   8   0   0  2   6  0  0   3  12
5      Aldrete   Mike  25  84 256 216  27  54  18   3  2  25  1  3  33  34
6    Alexander  Doyle  35  18  45  38   2   8   1   0  0   5  0  0   0   8
7     Allanson   Andy  24 101 324 293  30  66   7   3  1  29 10  1  14  36
8        Almon   Bill  33 102 230 196  29  43   7   2  7  27 11  4  30  38
9      Amelung     Ed  27   8  11  11   0   1   0   0  0   0  0  0   0   4
10    Andersen  Larry  33  48   7   6   0   0   0   0  0   0  0  0   0   3
11    Anderson   Dave  25  92 241 216  31  53   9   0  1  15  5  1  22  39
12    Anderson   Rick  29  15  12  11   1   1   0   0  0   0  0  0   0   4
13       Armas   Tony  32 121 453 425  40 112  21   4 11  58  0  3  24  77
14     Asadoor  Randy  23  15  60  55   9  20   5   0  0   7  1  2   3  13
15       Ashby   Alan  34 120 361 315  24  81  15   0  7  38  1  0  39  56
```

Q6. Add a new column batting average called BA. Batting average is computed by the number of hits (H) divided by the number of at bats (AB). Store the result in baseball.

A6.

```
127  baseball <- baseball %>%
128    mutate(BA = H / AB)
129  baseball
```

129:1 Assignmnet2 R Scri

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> baseball
          Last  First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS  BB  SO         BA
1        Acker    Jim  27  21  28  28   1   3   1   0  0   0  0  0   0  21 0.10714286
2       Adduci    Jim  26   3  13  11   2   1   1   0  0   0  0  0   1   2 0.09090909
3       Aguayo   Luis  27  62 146 133  17  28   6   1  4  13  1  1   8  26 0.21052632
4     Aguilera   Rick  24  32  57  51   4   8   0   0  2   6  0  0   3  12 0.15686275
5      Aldrete   Mike  25  84 256 216  27  54  18   3  2  25  1  3  33  34 0.25000000
6    Alexander  Doyle  35  18  45  38   2   8   1   0  0   5  0  0   0   8 0.21052632
7     Allanson   Andy  24 101 324 293  30  66   7   3  1  29 10  1  14  36 0.22525597
8        Almon   Bill  33 102 230 196  29  43   7   2  7  27 11  4  30  38 0.21938776
9      Amelung     Ed  27   8  11  11   0   1   0   0  0   0  0  0   0   4 0.09090909
10    Andersen  Larry  33  48   7   6   0   0   0   0  0   0  0  0   0   3 0.00000000
11    Anderson   Dave  25  92 241 216  31  53   9   0  1  15  5  1  22  39 0.24537037
12    Anderson   Rick  29  15  12  11   1   1   0   0  0   0  0  0   0   4 0.09090909
13       Armas   Tony  32 121 453 425  40 112  21   4 11  58  0  3  24  77 0.26352941
14     Asadoor  Randy  23  15  60  55   9  20   5   0  0   7  1  2   3  13 0.36363636
15       Ashby   Alan  34 120 361 315  24  81  15   0  7  38  1  0  39  56 0.25714286
```

Q7. Modify your new BA column so that the value is rounded to three (3) decimal places.

A7.

```
131  baseball$BA <- round(baseball$BA, 3)
132  baseball
133
134
```

132:1 Assignmnet2

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> baseball
          Last  First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS  BB  SO    BA
1        Acker    Jim  27  21  28  28   1   3   1   0  0   0  0  0   0  21 0.107
2       Adduci    Jim  26   3  13  11   2   1   1   0  0   0  0  0   1   2 0.091
3       Aguayo   Luis  27  62 146 133  17  28   6   1  4  13  1  1   8  26 0.211
4     Aguilera   Rick  24  32  57  51   4   8   0   0  2   6  0  0   3  12 0.157
5      Aldrete   Mike  25  84 256 216  27  54  18   3  2  25  1  3  33  34 0.250
6    Alexander  Doyle  35  18  45  38   2   8   1   0  0   5  0  0   0   8 0.211
7     Allanson   Andy  24 101 324 293  30  66   7   3  1  29 10  1  14  36 0.225
8        Almon   Bill  33 102 230 196  29  43   7   2  7  27 11  4  30  38 0.219
9      Amelung     Ed  27   8  11  11   0   1   0   0  0   0  0  0   0   4 0.091
10    Andersen  Larry  33  48   7   6   0   0   0   0  0   0  0  0   0   3 0.000
11    Anderson   Dave  25  92 241 216  31  53   9   0  1  15  5  1  22  39 0.245
12    Anderson   Rick  29  15  12  11   1   1   0   0  0   0  0  0   0   4 0.091
13       Armas   Tony  32 121 453 425  40 112  21   4 11  58  0  3  24  77 0.264
14     Asadoor  Randy  23  15  60  55   9  20   5   0  0   7  1  2   3  13 0.364
15       Ashby   Alan  34 120 361 315  24  81  15   0  7  38  1  0  39  56 0.257
```

Q8. On-base percentage (OBP) is arguably a better statistic than batting average. Create a column called OBP that computes this stat as (H + BB) / (AB + BB). Store the result in baseball.

A8.

```
134 baseball <- baseball %>%
135   mutate(OBP = (H + BB) / (AB + BB))
136 baseball
137
138
139
136:1  # Assignmnet2
```

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> baseball
          Last  First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS  BB  SO    BA        OBP
1        Acker    Jim  27  21  28  28   1   3   1   0  0   0  0  0   0  21 0.107 0.10714286
2       Adduci    Jim  26   3  13  11   2   1   1   0  0   0  0  0   1   2 0.091 0.16666667
3       Aguayo   Luis  27  62 146 133  17  28   6   1  4  13  1  1   8  26 0.211 0.25531915
4     Aguilera   Rick  24  32  57  51   4   8   0   0  2   6  0  0   3  12 0.157 0.20370370
5      Aldrete   Mike  25  84 256 216  27  54  18   3  2  25  1  3  33  34 0.250 0.34939759
6    Alexander  Doyle  35  18  45  38   2   8   1   0  0   5  0  0   0   8 0.211 0.21052632
7     Allanson   Andy  24 101 324 293  30  66   7   3  1  29 10  1  14  36 0.225 0.26058632
8        Almon   Bill  33 102 230 196  29  43   7   2  7  27 11  4  30  38 0.219 0.32300885
9      Amelung     Ed  27   8  11  11   0   1   0   0  0   0  0  0   0   4 0.091 0.09090909
10    Andersen  Larry  33  48   7   6   0   0   0   0  0   0  0  0   0   3 0.000 0.00000000
11    Anderson   Dave  25  92 241 216  31  53   9   0  1  15  5  1  22  39 0.245 0.31512605
12    Anderson   Rick  29  15  12  11   1   1   0   0  0   0  0  0   0   4 0.091 0.09090909
13       Armas   Tony  32 121 453 425  40 112  21   4 11  58  0  3  24  77 0.264 0.30289532
```

Q9. Modify your new OBP column so that the value is rounded to three (3) decimal

places.

A9.

```
138 baseball$OBP <- round(baseball$OBP, 3)
139 baseball
140
141
142
139:9  # Assignmnet2
```

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> baseball
          Last  First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS  BB  SO    BA    OBP
1        Acker    Jim  27  21  28  28   1   3   1   0  0   0  0  0   0  21 0.107 0.107
2       Adduci    Jim  26   3  13  11   2   1   1   0  0   0  0  0   1   2 0.091 0.167
3       Aguayo   Luis  27  62 146 133  17  28   6   1  4  13  1  1   8  26 0.211 0.255
4     Aguilera   Rick  24  32  57  51   4   8   0   0  2   6  0  0   3  12 0.157 0.204
5      Aldrete   Mike  25  84 256 216  27  54  18   3  2  25  1  3  33  34 0.250 0.349
6    Alexander  Doyle  35  18  45  38   2   8   1   0  0   5  0  0   0   8 0.211 0.211
7     Allanson   Andy  24 101 324 293  30  66   7   3  1  29 10  1  14  36 0.225 0.261
8        Almon   Bill  33 102 230 196  29  43   7   2  7  27 11  4  30  38 0.219 0.323
9      Amelung     Ed  27   8  11  11   0   1   0   0  0   0  0  0   0   4 0.091 0.091
10    Andersen  Larry  33  48   7   6   0   0   0   0  0   0  0  0   0   3 0.000 0.000
11    Anderson   Dave  25  92 241 216  31  53   9   0  1  15  5  1  22  39 0.245 0.315
12    Anderson   Rick  29  15  12  11   1   1   0   0  0   0  0  0   0   4 0.091 0.091
13       Armas   Tony  32 121 453 425  40 112  21   4 11  58  0  3  24  77 0.264 0.303
```

Q10. Determine the 10 players who struck out the most this season. Store these results as strikeout_artist.

A10.

```
141  strikeout_artist <- baseball %>%
142    arrange(desc(SO)) %>%
143    head(10)
144  strikeout_artist
```

144:1 # Assignmnet2 R

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> strikeout_artist
         Last  First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS BB  SO    BA   OBP
1  Incaviglia   Pete  22 153 606 540  82 135  21   2 30  88  3  2 55 185 0.250 0.319
2        Deer    Rob  25 134 546 466  75 108  17   3 33  86  5  2 72 179 0.232 0.335
3     Canseco   Jose  21 157 682 600  85 144  29   1 33 117 15  7 65 175 0.240 0.314
4     Presley    Jim  24 155 660 616  83 163  33   4 27 107  0  4 32 172 0.265 0.301
5   Tartabull  Danny  23 137 578 511  76 138  25   6 25  96  4  8 61 157 0.270 0.348
6     Balboni  Steve  29 138 562 512  54 117  25   1 29  88  0  0 43 146 0.229 0.288
7    Barfield  Jesse  26 158 671 589 107 170  35   2 40 108  8  8 69 146 0.289 0.363
8      Samuel   Juan  25 145 633 591  90 157  36  12 16  78 42 14 26 142 0.266 0.297
9      Murphy   Dale  30 160 692 614  89 163  29   7 29  83  7  7 75 141 0.265 0.345
10 Strawberry Darryl  24 136 562 475  76 123  27   5 27  93 28 12 72 141 0.259 0.356
>
```

Q11. Using a scatterplot (geom_point), plot the number of home runs (HRs) (the x-axis), versus the number of RBIs (the y-axis) per player.

A11.

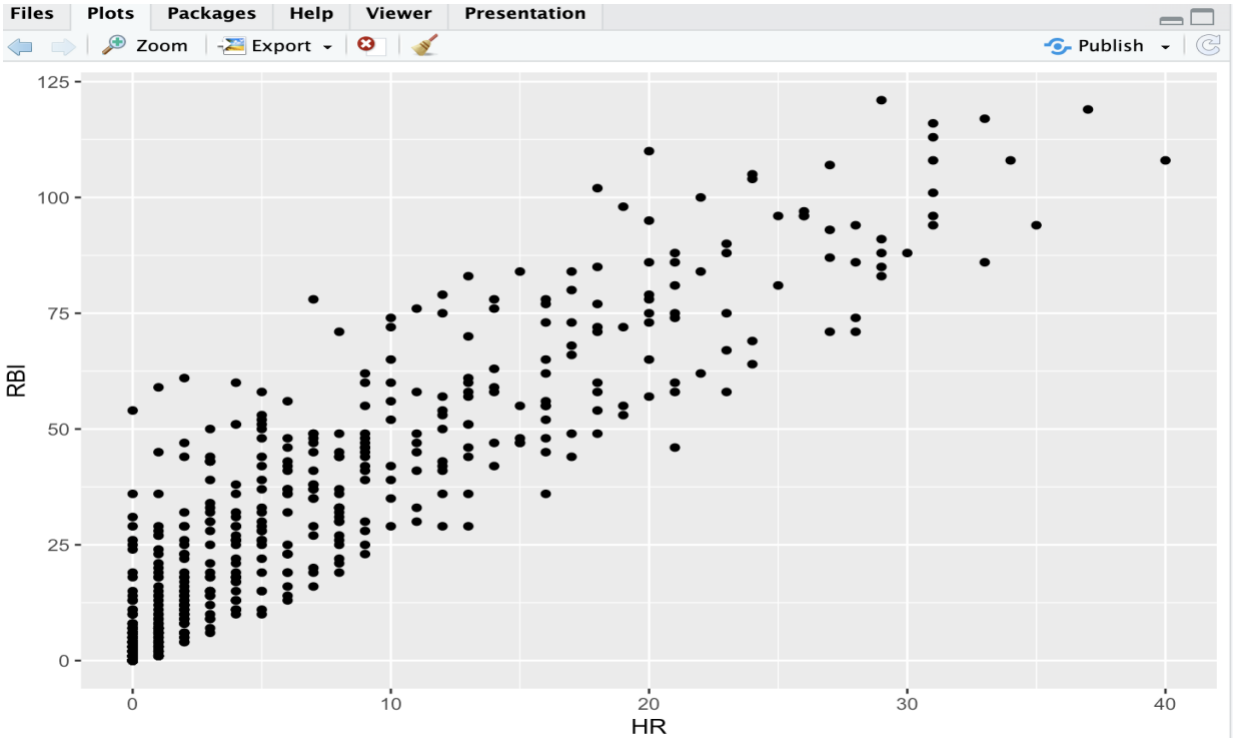
```
146  ggplot(data = baseball, aes(x = HR, y = RBI)) +
147    geom_point() +
148    labs(x = "HR", y = "RBI")
149
150
151
152
```

148:28 # Assignmnet2

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics

```
> ggplot(data = baseball, aes(x = HR, y = RBI)) +
+   geom_point() +
+   labs(x = "HR", y = "RBI")
>
```



Q12. To be eligible for end-of-season awards, a player must have either at least 300 at bats or appear in at least 100 games. Keep only the players who are eligible to be considered and store them in a variable called eligible_df.

A12.

```
150  eligible_df <- baseball %>%
151    filter(AB >= 300 | G >= 100)
152  eligible_df
153
154
155
156
152:1  Assignmnet2
```

Console | Terminal | Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

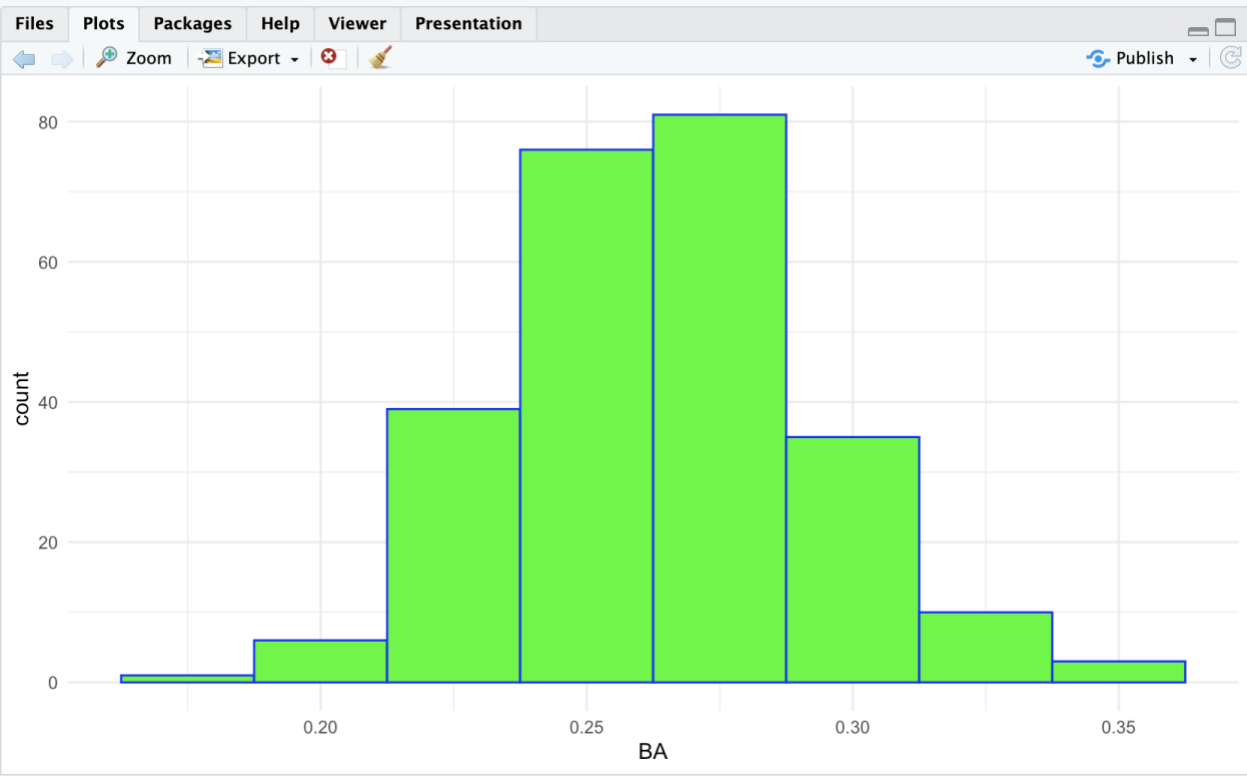
```
> eligible_df
          Last    First Age   G  PA  AB   R   H X2B X3B HR RBI  SB CS  BB  SO    BA   OBP
1    Allanson     Andy  24 101 324 293  30  66   7   3  1  29  10  1  14  36 0.225 0.261
2       Almon     Bill  33 102 230 196  29  43   7   2  7  27  11  4  30  38 0.219 0.323
3       Armas     Tony  32 121 453 425  40 112  21   4 11  58   0  3  24  77 0.264 0.303
4       Ashby     Alan  34 120 361 315  24  81  15   0  7  38   1  0  39  56 0.257 0.339
5     Backman    Wally  26 124 440 387  67 124  18   2  1  27  13  7  36  32 0.320 0.378
6      Baines   Harold  27 145 618 570  72 169  29   2 21  88   2  1  38  89 0.296 0.340
7      Balboni    Steve  29 138 562 512  54 117  25   1 29  88   0  0  43 146 0.229 0.288
8     Barfield    Jesse  26 158 671 589 107 170  35   2 40 108   8  8  69 146 0.289 0.363
9     Barrett    Marty  28 158 713 625  94 179  39   4  4  60  15  7  65  31 0.286 0.354
10       Bass    Kevin  27 157 640 591  83 184  33   5 20  79  22 13  38  72 0.311 0.353
11     Baylor      Don  37 160 687 585  93 139  23   1 31  94   3  5  62 111 0.238 0.311
12       Bell    Buddy  34 155 655 568  89 158  29   3 20  75   2  8  73  49 0.278 0.360
13       Bell   George  26 159 690 641 101 198  38   6 31 108   7  8  41  62 0.309 0.350
14   Belliard   Rafael  24 117 350 309  33  72   5   2  0  31  12  2  26  54 0.233 0.293
15   Beniquez     Juan  36 113 395 343  48 103  15   0  6  36   2  3  40  49 0.300 0.373
16  Bernazard     Tony  29 146 636 562  88 169  28   4 17  73  17  8  53  77 0.301 0.361
17 Biancalana    Buddy  26 100 209 190  24  46   4   4  2   8   5  1  15  50 0.242 0.298
```

Q13. For eligible players, create a histogram of batting average. Use a binwidth of .025 in your graph. The graph should be drawn in blue and filled in green.

A13.

```
154 ggplot(data = eligible_df, aes(x = BA)) +
155   geom_histogram(binwidth = 0.025, fill = "green", color = "blue") +
156   labs(x = "BA", y = "count") +
157   theme_minimal()
158
159
160
161
159:1  # Assignmnet2
```

```
Console  Terminal  Background Jobs
R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/
> ggplot(data = eligible_df, aes(x = BA)) +
+   geom_histogram(binwidth = 0.025, fill = "green", color = "blue") +
+   labs(x = "BA", y = "count") +
+   theme_minimal()
>
```



Q14. Use the following code to create a ranking column of eligible players with regard to home runs (HRs). Store the result in eligible_df.

A14.

```
159  eligible_df <- eligible_df |>
160    mutate(RankHR =rank(-1 * HR, ties.method = "min"))
161  eligible_df
162
163
164
165
```

161:12 Assignmnet2

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> eligible_df
         Last   First Age   G  PA  AB   R   H X2B X3B HR RBI  SB CS  BB  SO    BA   OBP RankHR
1    Allanson    Andy  24 101 324 293  30  66   7   3  1  29  10  1  14  36 0.225 0.261    231
2       Almon    Bill  33 102 230 196  29  43   7   2  7  27  11  4  30  38 0.219 0.323    160
3       Armas    Tony  32 121 453 425  40 112  21   4 11  58   0  3  24  77 0.264 0.303    121
4       Ashby    Alan  34 120 361 315  24  81  15   0  7  38   1  0  39  56 0.257 0.339    160
5     Backman   Wally  26 124 440 387  67 124  18   2  1  27  13  7  36  32 0.320 0.378    231
6      Baines  Harold  27 145 618 570  72 169  29   2 21  88   2  1  38  89 0.296 0.340     44
7     Balboni   Steve  29 138 562 512  54 117  25   1 29  88   0  0  43 146 0.229 0.288     14
8    Barfield   Jesse  26 158 671 589 107 170  35   2 40 108   8  8  69 146 0.289 0.363      1
9     Barrett   Marty  28 158 713 625  94 179  39   4  4  60  15  7  65  31 0.286 0.354    196
10       Bass   Kevin  27 157 640 591  83 184  33   5 20  79  22 13  38  72 0.311 0.353     52
11     Baylor     Don  37 160 687 585  93 139  23   1 31  94   3  5  62 111 0.238 0.311      7
12       Bell   Buddy  34 155 655 568  89 158  29   3 20  75   2  8  73  49 0.278 0.360     52
13       Bell  George  26 159 690 641 101 198  38   6 31 108   7  8  41  62 0.309 0.350      7
14   Belliard  Rafael  24 117 350 309  33  72   5   2  0  31  12  2  26  54 0.233 0.293    243
15   Beniquez    Juan  36 113 395 343  48 103  15   0  6  36   2  3  40  49 0.300 0.373    172
16  Bernazard    Tony  29 146 636 562  88 169  28   4 17  73  17  8  53  77 0.301 0.361     74
17 Biancalana   Buddy  26 100 209 190  24  46   4   4  2   8   5  1  15  50 0.242 0.298    219
```

Q15. Repeat the prior step to create rankings for both runs batted in (RBI) and on-base percentage (OBP). Store the result in eligible_df.

A15.

```
163 eligible_df <- eligible_df |>
164   mutate(RankRBI =rank(-1 * RBI, ties.method = "min")) %>%
165   mutate(RankOBP =rank(-1 * OBP, ties.method = "min"))
166 eligible_df
167
```

167:1 Assignmnet2

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> eligible_df
        Last   First Age   G  PA  AB   R   H X2B X3B HR RBI  SB CS  BB  SO    BA   OBP RankHR RankRBI RankOBP
1  Allanson    Andy  24 101 324 293  30  66   7   3  1  29  10  1  14  36 0.225 0.261    231     215     246
2     Almon    Bill  33 102 230 196  29  43   7   2  7  27  11  4  30  38 0.219 0.323    160     224     152
3     Armas    Tony  32 121 453 425  40 112  21   4 11  58   0  3  24  77 0.264 0.303    121      98     196
4     Ashby    Alan  34 120 361 315  24  81  15   0  7  38   1  0  39  56 0.257 0.339    160     185     108
5   Backman   Wally  26 124 440 387  67 124  18   2  1  27  13  7  36  32 0.320 0.378    231     224      24
6    Baines  Harold  27 145 618 570  72 169  29   2 21  88   2  1  38  89 0.296 0.340     44      29     103
7   Balboni   Steve  29 138 562 512  54 117  25   1 29  88   0  0  43 146 0.229 0.288     14      29     225
8  Barfield   Jesse  26 158 671 589 107 170  35   2 40 108   8  8  69 146 0.289 0.363      1       7      45
```

Q16. Create a TotalRank column that is the sum of the prior three (3) ranks. If a player was ranked first in HR, RBI, and OBP, then their total rank would be 3. Store the result in eligible_df.

A16.

```
169 eligible_df <- eligible_df %>%
170   mutate(TotalRank = RankHR + RankRBI + RankOBP)
171 eligible_df
172
```

169:31 Assignmnet2

Console Terminal Background Jobs

R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/

```
> eligible_df
        Last   First Age   G  PA  AB   R   H X2B X3B HR RBI  SB CS  BB  SO    BA   OBP RankHR RankRBI RankOBP TotalRank
1  Allanson    Andy  24 101 324 293  30  66   7   3  1  29  10  1  14  36 0.225 0.261    231     215     246       692
2     Almon    Bill  33 102 230 196  29  43   7   2  7  27  11  4  30  38 0.219 0.323    160     224     152       536
3     Armas    Tony  32 121 453 425  40 112  21   4 11  58   0  3  24  77 0.264 0.303    121      98     196       415
4     Ashby    Alan  34 120 361 315  24  81  15   0  7  38   1  0  39  56 0.257 0.339    160     185     108       453
5   Backman   Wally  26 124 440 387  67 124  18   2  1  27  13  7  36  32 0.320 0.378    231     224      24       479
6    Baines  Harold  27 145 618 570  72 169  29   2 21  88   2  1  38  89 0.296 0.340     44      29     103       176
7   Balboni   Steve  29 138 562 512  54 117  25   1 29  88   0  0  43 146 0.229 0.288     14      29     225       268
8  Barfield   Jesse  26 158 671 589 107 170  35   2 40 108   8  8  69 146 0.289 0.363      1       7      45        53
```

Q17. Arrange the data in ascending order by TotalRank and store the twenty (20) lowest TotalRank scores in a variable called mvp_candidates.

A17.

```
173  mvp_candidates <- eligible_df %>%
174    arrange(TotalRank) %>%
175    head(20)
176  mvp_candidates
176:15  # Assignmnet2 ⇕

Console  Terminal ×  Background Jobs ×
R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/
> mvp_candidates
        Last   First Age   G  PA  AB   R   H X2B X3B HR RBI SB CS BB  SO    BA   OBP RankHR RankRBI RankOBP TotalRank
1   Mattingly     Don  25 162 742 677 117 238  53   2 31 113  0  0 53  35 0.352 0.399      7       5       8        20
2     Schmidt    Mike  36 160 657 552  97 160  29   1 37 119  1  2 89  84 0.290 0.388      2       2      16        20
3    Barfield   Jesse  26 158 671 589 107 170  35   2 40 108  8  8 69 146 0.289 0.363      1       7      45        53
4       Evans  Dwight  34 152 640 529  86 137  33   2 26  97  3  3 97 117 0.259 0.374     27      17      30        74
5     Puckett   Kirby  26 161 723 680 119 223  37   6 31  96 20 12 34  99 0.328 0.360      7      18      50        75
6        Rice     Jim  33 157 693 618  98 200  39   2 20 110  0  1 62  78 0.324 0.385     52       6      18        76
7     O'Brien    Pete  28 156 641 551  86 160  23   3 23  90  4  4 87  66 0.290 0.387     36      28      17        81
8        Bell  George  26 159 690 641 101 198  38   6 31 108  7  8 41  62 0.309 0.350      7       7      74        88
9  McReynolds   Kevin  26 158 641 560  89 161  31   6 26  96  8  6 66  83 0.288 0.363     27      18      45        90
10     Gibson    Kirk  29 119 521 441  84 118  11   2 28  86 34  6 68 107 0.268 0.365     19      34      41        94
11     Gaetti    Gary  27 157 661 596  91 171  34   1 34 108 14 15 52 108 0.287 0.344      4       7      86        97
12      Hayes     Von  27 158 690 610 107 186  46   2 19  98 24 12 74  77 0.305 0.380     61      16      21        98
13    Downing   Brian  35 152 631 513  90 137  27   4 20  95  4  4 90  84 0.267 0.376     52      22      28       102
14 Strawberry  Darryl  24 136 562 475  76 123  27   5 27  93 28 12 72 141 0.259 0.356     23      26      57       106
15      Evans Darrell  39 151 601 507  78 122  15   0 29  85  3  2 91 105 0.241 0.356     14      38      57       109
16      Hrbek    Kent  26 149 634 550  85 147  27   1 29  91  2  2 71  81 0.267 0.351     14      27      71       112
17      Davis    Eric  24 132 487 415  97 115  15   3 27  71 80 11 68 100 0.277 0.379     23      71      22       116
18   Winfield    Dave  34 154 652 565  90 148  31   5 24 104  6  5 77 106 0.262 0.350     32      12      74       118
19    Parrish   Larry  32 129 524 464  67 128  22   1 28  94  3  1 52 114 0.276 0.349     19      23      77       119
20     Murray   Eddie  30 137 578 495  61 151  25   1 17  84  3  0 78  49 0.305 0.400     74      40       6       120
```

Q18. Create a variable called mvp_candidates_abbreviated with the First, Last, RankHR, RankRBI, and RankOBP selected from mvp_candidates.

A18.

```
178  mvp_candidates_abbreviated <- mvp_candidates %>%
179    select(First, Last, RankHR, RankRBI, RankOBP)
180  mvp_candidates_abbreviated
180:1  # Assignmnet2 ⇕

Console  Terminal ×  Background Jobs ×
R 4.3.1 · ~/Documents/Courses/Quarter1/ALY600-IntroToAnalytics/Assignments/Week2/Rohilla_Project2/
> mvp_candidates_abbreviated
      First        Last RankHR RankRBI RankOBP
1       Don   Mattingly      7       5       8
2      Mike     Schmidt      2       2      16
3     Jesse    Barfield      1       7      45
4    Dwight       Evans     27      17      30
5     Kirby     Puckett      7      18      50
6       Jim        Rice     52       6      18
7      Pete     O'Brien     36      28      17
8    George        Bell      7       7      74
9     Kevin  McReynolds     27      18      45
10     Kirk      Gibson     19      34      41
11     Gary      Gaetti      4       7      86
12      Von       Hayes     61      16      21
13    Brian     Downing     52      22      28
14   Darryl  Strawberry     23      26      57
15 Darrell       Evans     14      38      57
16     Kent       Hrbek     14      27      71
17     Eric       Davis     23      71      22
18     Dave    Winfield     32      12      74
19    Larry     Parrish     19      23      77
20    Eddie      Murray     74      40       6
>
```

Q19. Make a recommendation for the league most valuable player (MVP). Keep in mind that the dataset completely ignores pitchers. You can decide whether a pitcher should be eligible for the MVP. Base your decision on the data you have analyzed.

A19.

Recommendation

- Performance on offense: The MVP award generally places a strong emphasis on an offense, and position players primarily contribute to an offense by hitting, running bases, and playing defense. Key offensive statistics to consider include batting average, on-base percentage (OBP), slugging percentage (SLG), home runs (HR), runs batted in (RBI), and runs scored (R).

  The Best Pitcher Award, one of the prizes specifically given to pitchers, honors their contributions to the sport.
- Consistency and Contribution: Eligible players must have contributed consistently and significantly to their teams throughout the season in order to meet the requirements of 300 AB or 100 Games. This fits with the concept of the MVP award, which is given to athletes who significantly contribute to their team's victory.
- Data Analysis: Analyzed the information with a focus on position players' batting statistics, such as average, on-base percentage, home runs, and runs batted in. Adding pitchers' statistics and factors into the MVP selection could make the evaluation process more difficult. Player who consistently performed well from the beginning to the end of the season may have a stronger case. Compare the statistical accomplishments of MVP

contenders to those of other league players while taking into account their roles and positions.

- The awarding for the MVP to position players is a custom in baseball, should uphold fairness and honesty in the voting process.

##########**Citations**##########

- https://www.baseball-reference.com/leagues/majors/1986-standard-batting.shtml
- https://northeastern.instructure.com/courses/160343/pages/module-2-%7C-resources?module_item_id=9214302
- https://northeastern.instructure.com/courses/160343/pages/module-2-%7C-resources?module_item_id=9214302
- https://www.geeksforgeeks.org/data-visualization-with-r-and-ggplot2/
- https://northeastern.instructure.com/courses/160343/pages/module-2-%7C-resources?module_item_id=9214302
- https://stackoverflow.com/questions/38480183/how-to-plot-histogram-in-r