Project 2

Justin Williams

2024-10-23

Let's load some data and packages!

```
library(tidyverse)
library(dplyr)

# Load the data from the CSV file into a data frame
HRData_New <- read_csv("/Users/justinwilliams/Code/9050advresearch/Project 2/HRData_New.csv")
```

```
## Rows: 311 Columns: 38
## -- Column specification ------------------------------------------------------------------------
## Delimiter: ","
## chr  (17): Employee_Name, Position, State, Zip, Sex, MaritalDesc, CitizenDes...
## dbl  (18): EmpID, MarriedID, MaritalStatusID, GenderID, EmpStatusID, DeptID,...
## date  (3): DOB, DateofHire, LastPerformanceReview_Date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
hrdata_df <- data.frame(HRData_New)
```

1. Report the descriptive statistics along with the frequency distribution and provide a detailed interpretation of how you would characterize the salary variable.

Summary statistics

```
## Summary statistics
summary(hrdata_df$Salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   45046   55502   62810   69021   72036  250000
```

```
## Mean
mean_salary <- mean(hrdata_df$Salary, na.rm = TRUE)
print(mean_salary)
```

```
## [1] 69020.68
```

```r
## Median
median_salary <- median(hrdata_df$Salary, na.rm = TRUE)
print(median_salary)
```

```
## [1] 62810
```

```r
## Mode
mode_salary <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
mode_salary_value <- mode_salary(hrdata_df$Salary)
print(mode_salary_value)
```

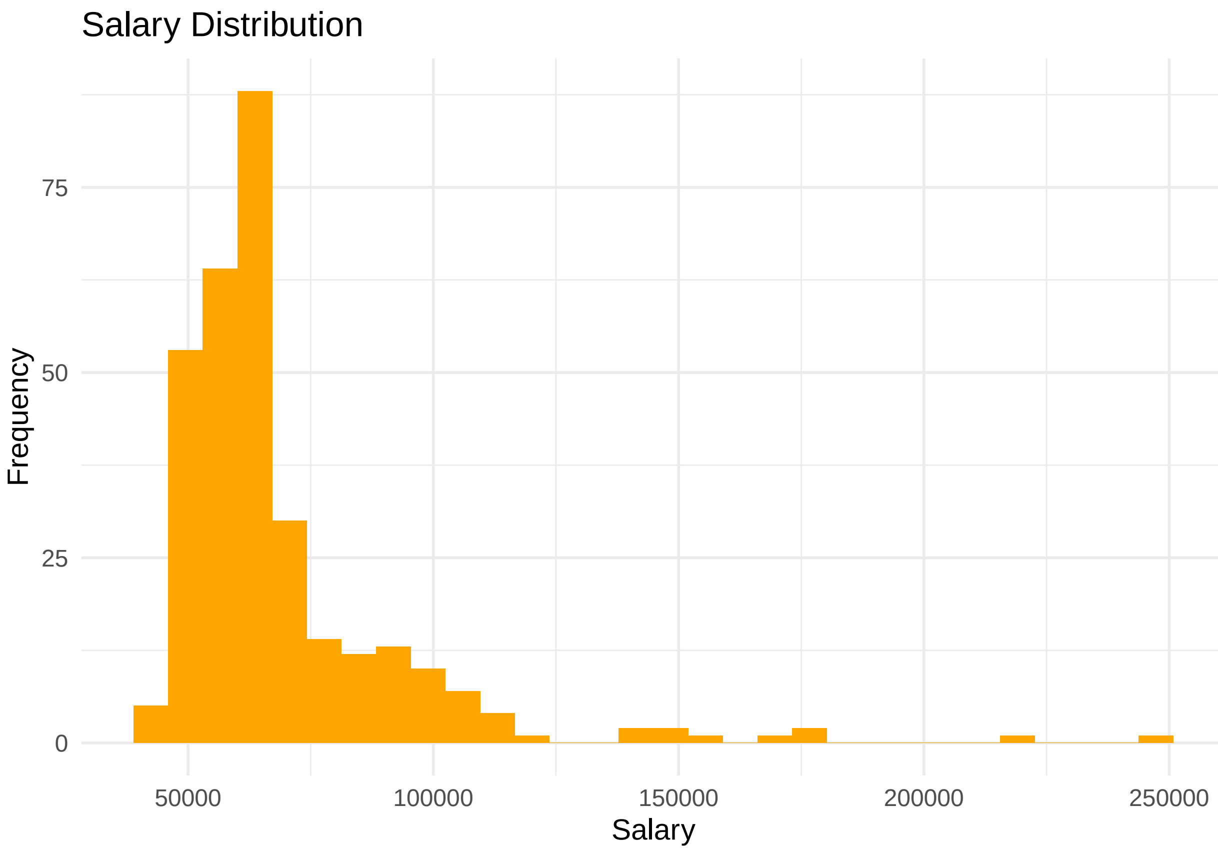
```
## [1] 57815
```

```r
## Frequency table
salary_table <- table(hrdata_df$Salary)
print(salary_table)
```

```
## 
## 45046 45069 45115 45395 45433 45998 46120 46335 46428 46430 46654 
##     1     1     1     1     1     1     1     1     1     1     1 
## 46664 46738 46799 46837 46998 47001 47211 47414 47434 47750 47837 
##     1     1     1     1     1     1     1     1     1     1     1 
## 47961 48285 48413 48495 48513 48888 49256 49773 49920 50178 50274 
##     1     1     1     1     1     1     1     1     1     1     1 
## 50373 50428 50470 50482 50750 50825 50923 51044 51259 51337 51505 
##     1     1     1     1     1     1     1     1     1     1     1 
## 51777 51908 51920 52057 52087 52177 52249 52505 52599 52624 52674 
##     1     1     1     1     1     1     1     1     1     1     1 
## 52788 52846 52984 53018 53060 53171 53180 53189 53250 53366 53492 
##     1     1     1     1     1     1     1     1     1     1     1 
## 53564 54005 54132 54237 54285 54670 54828 54933 55000 55140 55315 
##     1     1     1     1     1     1     1     1     1     1     1 
## 55425 55578 55688 55722 55800 55875 55965 56147 56149 56294 56339 
##     1     1     1     1     1     1     1     1     1     1     1 
## 56847 56991 57568 57575 57583 57748 57815 57834 57859 57954 57975 
##     1     1     1     1     1     1     2     1     1     1     1 
## 58062 58207 58273 58275 58370 58371 58523 58530 58709 58939 59026 
##     1     1     1     1     1     1     1     1     1     1     1 
## 59124 59144 59231 59238 59365 59369 59370 59472 59728 59892 60070 
##     1     1     1     1     1     1     1     1     1     1     1 
## 60120 60270 60340 60380 60436 60446 60627 60656 60724 60754 61154 
##     1     1     1     1     1     1     1     1     1     1     1 
## 61242 61349 61355 61422 61555 61568 61584 61656 61729 61809 61844 
##     2     1     1     1     1     1     1     1     1     1     1 
## 61962 62061 62065 62068 62162 62385 62425 62506 62514 62659 62810 
##     1     1     1     1     1     1     1     1     1     1     1 
## 62910 62957 63000 63003 63025 63051 63108 63291 63322 63353 63381
```

```
##       1       1       1       1       2       1       1       1       1       1       1
##   63430   63450   63478   63515   63676   63682   63695   63763   63813   63878   63973
##       1       1       1       1       1       1       1       1       1       1       1
##   64021   64057   64066   64246   64375   64397   64520   64724   64738   64786   64816
##       1       1       1       1       1       1       1       1       1       1       1
##   64919   64955   64971   64991   64995   65288   65310   65707   65714   65729   65893
##       1       1       1       1       1       1       1       1       1       1       1
##   65902   66074   66149   66441   66541   66593   66738   66808   66825   67176   67237
##       1       1       1       1       1       1       1       1       1       1       1
##   67251   68051   68099   68182   68407   68678   68829   68898   68999   69340   70131
##       1       1       1       1       1       1       1       1       1       1       1
##   70187   70468   70507   70545   70621   71339   71707   71776   71860   71966   72106
##       1       1       1       1       1       1       1       1       1       1       1
##   72202   72460   72609   72640   72992   73330   74226   74241   74312   74326   74417
##       1       1       1       1       1       1       1       1       1       1       1
##   74669   74679   74813   75188   75281   76029   77692   77915   80512   81584   82758
##       1       1       1       1       1       1       1       1       1       1       1
##   83082   83363   83552   83667   84903   85028   86214   87565   87826   87921   88527
##       1       1       1       1       1       1       1       1       1       1       1
##   88976   89292   89883   90100   92328   92329   92989   93046   93093   93206   93396
##       1       1       1       1       1       1       1       1       1       1       1
##   93554   95660   95920   96820   97999   99020   99280   99351  100031  100416  101199
##       1       1       1       1       1       1       1       1       1       1       1
##  103613  104437  105688  105700  106367  107226  108987  110000  110929  113999  114800
##       1       1       1       1       1       1       1       1       1       1       1
##  120000  138888  140920  148999  150290  157000  170500  178000  180000  220450  250000
##       1       1       1       1       1       1       1       1       1       1       1
```

```r
## Plot
hrdata_df %>%
  ggplot(aes(x = Salary)) +
  geom_histogram(fill = "orange", bins = 30) +
  labs(title = "Salary Distribution", x = "Salary", y = "Frequency") +
  theme_minimal()
```



Interpretation ### Summary Interpretation

```
## The mean salary is 69020.68
```

```
## The median salary is 62810
```

```
## The mode salary is 57815
```

```
## The frequency distribution of salaries is shown in the bar plot above.
```

Detailed Interpretation

- The mean salary is $69,020.68, which indicates the average salary of the employees. The range is large as well as a large variation in salaries as the max salary is $250,00, which is a high end outlier. The mean salary is higher than the median salary, meaning that the distribution is right-skewed suggesting that most employees earn less than the mean.
- The median salary is $62,810, which is the middle value when the salaries are sorted in ascending order. The median gives us a more accurate view of what the typical employee is earning, due to the right-skewness of the distribution which impacts the mean. This could be due to company factors such as the majority of positions in the comapny being more junior.
- The mode salary is $57,815, which is the most frequently occurring salary in the dataset. The mode being less than both the mean and median again shows the right-skewness of the distribution.
- The frequency distribution bar plot shows the distribution of salaries across different ranges. This helps in visualizing how salaries are spread out among the employees.

Overall, the salary distribution is spread across a wide range of values with the majority of employees falling below the mean. A small number of employees earn significantly higher salaries than the rest, as indicated by the maximum salary of $250,000. The median may be a better measure of central tendency in this case due to the presence of large outliers.

2. Which employee (ID number) has the largest z-score on Salary and what is the z-score for this person? Which employee (ID number) has the smallest z-score on Salary and what is the z-score for this person?

```r
# Calculate z-scores for Salary
salary_zscore <- scale(hrdata_df$Salary, center = TRUE, scale = TRUE)
print(salary_zscore)
```

```
##                [,1]
##   [1,]  0.1438711829
##   [2,] -0.4552550056
##   [3,] -0.2429054768
##   [4,] -0.8580910456
##   [5,]  1.5886986494
##   [6,]  0.2075124400
##   [7,] -0.2764950222
##   [8,] -0.6854527072
##   [9,] -0.3298010346
##  [10,]  6.0194578288
##  [11,] -0.8857577008
##  [12,]  0.9264877168
##  [13,] -0.1837958269
##  [14,] -0.4172928566
##  [15,]  4.3320303671
##  [16,]  0.9752223710
##  [17,]  0.3535573987
##  [18,] -0.4454365231
##  [19,]  4.0338983067
##  [20,] -0.2252958097
##  [21,] -0.8588860645
##  [22,] -0.7651135937
##  [23,]  0.0441360710
##  [24,]  0.9527630891
##  [25,]  0.1367160134
##  [26,] -0.2589648571
##  [27,] -0.3325040987
##  [28,]  2.7772915477
##  [29,] -0.7412630289
##  [30,] -0.2621846833
##  [31,] -0.3628738179
##  [32,] -0.4399111423
##  [33,]  0.0590824249
##  [34,] -0.8818223576
##  [35,]  0.1712993324
```

```
##  [36,] -0.2265280889
##  [37,] -0.6374733209
##  [38,] -0.1789064611
##  [39,] -0.6797683226
##  [40,]  0.1128654486
##  [41,] -0.0076196547
##  [42,] -0.2117009878
##  [43,]  0.8058038588
##  [44,]  0.2488534191
##  [45,]  0.2785871232
##  [46,] -0.7145901472
##  [47,] -0.7390767271
##  [48,] -0.5517702913
##  [49,] -0.4170146000
##  [50,] -0.1707972691
##  [51,] -0.8242629945
##  [52,] -0.9103237826
##  [53,] -0.5918392402
##  [54,] -0.3336568760
##  [55,] -0.3972981331
##  [56,] -0.4546587415
##  [57,] -0.6743224436
##  [58,] -0.9502734787
##  [59,] -0.2070103767
##  [60,] -0.3412493058
##  [61,] -0.4446812552
##  [62,] -0.3838225640
##  [63,] -0.8159152968
##  [64,] -0.3557981503
##  [65,] -0.6296821364
##  [66,] -0.6565140219
##  [67,] -0.2927531573
##  [68,] -0.5299867754
##  [69,] -0.7232956034
##  [70,] -0.5286352434
##  [71,] -0.7194000111
##  [72,] -0.4233349997
##  [73,] -0.0243945520
##  [74,] -0.1897982191
##  [75,] -0.0366378419
##  [76,] -0.5448138766
##  [77,] -0.2637747210
##  [78,]  0.1170790484
##  [79,]  0.7513053182
##  [80,]  1.2056585782
##  [81,]  1.4845511829
##  [82,]  1.2326892183
##  [83,]  0.9265274678
##  [84,]  1.4078318661
##  [85,]  0.9689417222
##  [86,]  3.2305317813
##  [87,] -0.2124562557
##  [88,] -0.1601837678
##  [89,]  7.1940981467
```

```
##  [90,] -0.1314438371
##  [91,] -0.6731299154
##  [92,]  0.5460712078
##  [93,]  0.1426389037
##  [94,] -0.2241827834
##  [95,] -0.2044265655
##  [96,] -0.0709031534
##  [97,] -0.6695523306
##  [98,] -0.2410371825
##  [99,] -0.1308475730
## [100,] -0.4271510901
## [101,] -0.3092100470
## [102,]  1.2479933308
## [103,]  0.0575321382
## [104,] -0.0907388732
## [105,]  0.2451565815
## [106,] -0.1171334982
## [107,] -0.2089979238
## [108,]  1.6658949775
## [109,]  0.1067835546
## [110,]  1.4575602937
## [111,] -0.6666902629
## [112,]  1.1519153054
## [113,]  0.9613890434
## [114,] -0.8420714163
## [115,] -0.6429589509
## [116,]  0.5822048135
## [117,] -0.2383341185
## [118,]  1.7879303676
## [119,] -0.0965027597
## [120,]  0.2145086057
## [121,] -0.2373005940
## [122,] -0.6962649633
## [123,] -0.5041089126
## [124,] -0.2852799803
## [125,] -0.5641725850
## [126,]  0.6834504612
## [127,]  1.5186972416
## [128,] -0.2528829631
## [129,] -0.8754224561
## [130,]  0.2245258430
## [131,]  0.5701205273
## [132,] -0.5116615915
## [133,]  0.0636140322
## [134,]  0.9550288928
## [135,] -0.1243284186
## [136,] -0.1239706602
## [137,] -0.2214399684
## [138,] -0.3127081298
## [139,] -0.6802453339
## [140,] -0.2866712632
## [141,] -0.8191748740
## [142,] -0.3836238093
## [143,] -0.5058977050
```

```
## [144,]  0.7932425613
## [145,] -0.2468805709
## [146,]  0.1264602706
## [147,] -0.2392086392
## [148,] -0.2188561572
## [149,]  2.0264757668
## [150,]  0.3446929388
## [151,] -0.6527774334
## [152,] -0.2898513385
## [153,] -0.5573354231
## [154,] -0.3434753585
## [155,]  1.2791183179
## [156,]  1.4580373050
## [157,] -0.4007564650
## [158,] -0.2122177500
## [159,] -0.7029431214
## [160,] -0.0333385138
## [161,] -0.4233747507
## [162,]  0.8292966652
## [163,] -0.5189757647
## [164,] -0.8753032032
## [165,]  0.0921552081
## [166,] -0.5968876098
## [167,]  0.0605929607
## [168,] -0.1671401825
## [169,] -0.5117410934
## [170,] -0.9376326793
## [171,] -0.9151733975
## [172,]  0.6313369771
## [173,]  0.8379226194
## [174,] -0.3408517963
## [175,]  0.2103347569
## [176,] -0.2006502261
## [177,] -0.6172798427
## [178,]  0.7475289787
## [179,] -0.7262769240
## [180,]  0.7371539830
## [181,] -0.2726391809
## [182,] -0.7592702053
## [183,] -0.2763757694
## [184,] -0.1483777382
## [185,] -0.8886992705
## [186,] -0.6517836598
## [187,] -0.8833328934
## [188,]  0.2108912701
## [189,] -0.4480998362
## [190,] -0.0985698086
## [191,] -0.6452645055
## [192,] -0.7369301763
## [193,]  0.5776334553
## [194,]  1.0589378535
## [195,] -0.2203269421
## [196,] -0.1616148016
## [197,]  1.1050489455
```

```
## [198,]  2.8580654606
## [199,]  1.3750770904
## [200,] -0.0879563073
## [201,]  0.0126930763
## [202,] -0.2277603681
## [203,] -0.1846703476
## [204,] -0.4356180406
## [205,] -0.4436874817
## [206,] -0.2766540260
## [207,] -0.9017773302
## [208,] -0.8890967799
## [209,] -0.3888709335
## [210,] -0.5876653914
## [211,] -0.2586468496
## [212,]  0.6363058447
## [213,] -0.4298541541
## [214,] -0.1600247640
## [215,] -0.7374071876
## [216,] -0.4549767491
## [217,]  0.2069161759
## [218,] -0.1141521776
## [219,] -0.9391432151
## [220,] -0.0136220469
## [221,] -0.3286085064
## [222,]  3.1792133160
## [223,]  0.1226441802
## [224,] -0.5343593790
## [225,] -0.3795692133
## [226,] -0.1630458355
## [227,] -0.3092100470
## [228,]  0.2249233524
## [229,]  0.7753943886
## [230,] -0.1609787866
## [231,] -0.1475032175
## [232,]  0.4994036026
## [233,] -0.4390763725
## [234,]  1.1925010166
## [235,] -0.5857573462
## [236,] -0.9521020220
## [237,] -0.0872805413
## [238,] -0.2393278921
## [239,]  1.0692730983
## [240,] -0.1683327107
## [241,] -0.3538106032
## [242,] -0.8669555056
## [243,] -0.6361217889
## [244,] -0.0008619947
## [245,]  1.6289663530
## [246,]  1.8197708716
## [247,] -0.8002931769
## [248,] -0.5404412730
## [249,] -0.2805893692
## [250,] -0.7490144625
## [251,] -0.1702407559
```

```
## [252,] -0.5704532338
## [253,] -0.5225533494
## [254,] -0.2222349873
## [255,] -0.2967680024
## [256,] -0.7856648304
## [257,] -0.6300398949
## [258,] -0.0703466402
## [259,]  0.9568971870
## [260,] -0.8980804927
## [261,] -0.2350347904
## [262,] -0.8980009908
## [263,]  0.1095263695
## [264,] -0.3693929723
## [265,] -0.4098991815
## [266,] -0.3047181906
## [267,] -0.4454365231
## [268,] -0.0733279608
## [269,] -0.3836635602
## [270,]  0.2302499785
## [271,] -0.9530162937
## [272,]  4.4115322498
## [273,] -0.1317220937
## [274,]  0.4567905935
## [275,] -0.1969533885
## [276,] -0.7060436948
## [277,] -0.6268995705
## [278,] -0.8371422995
## [279,] -0.3049566963
## [280,] -0.3450653961
## [281,] -0.6344522494
## [282,] -0.0385458871
## [283,] -0.5599987362
## [284,] -0.2956152251
## [285,] -0.3020548776
## [286,] -0.6144177749
## [287,] -0.2383341185
## [288,]  3.4972605980
## [289,]  0.5589505128
## [290,]  1.2028362613
## [291,]  0.1578635143
## [292,] -0.3926075220
## [293,] -0.7451983721
## [294,] -0.1025449027
## [295,] -0.8455297482
## [296,] -0.3934025409
## [297,] -0.3478479620
## [298,] -0.5255346700
## [299,] -0.4839154344
## [300,] -0.0048768398
## [301,] -0.8151997799
## [302,] -0.1987421809
## [303,] -0.6497961128
## [304,] -0.3891491901
## [305,]  0.0463621237
```

```
## [306,] -0.2962512401
## [307,] -0.4272305920
## [308,] -0.1973111470
## [309,] -0.6222884613
## [310,] -0.6293243780
## [311,] -0.4781912988
## attr(,"scaled:center")
## [1] 69020.68
## attr(,"scaled:scale")
## [1] 25156.64
```

```r
# Find the index of the employee with the largest z-score
max_z_score_index <- which.max(salary_zscore)

# Retrieve the employee's EmpID and the corresponding z-score
max_z_score_employee <- hrdata_df[max_z_score_index, c("EmpID", "Salary")]

# Print the result
print(max_z_score_employee)
```

```
##    EmpID Salary
## 89 10089 250000
```

```r
# Find the index of the employee with the smallest z-score
min_z_score_index <- which.min(salary_zscore)

# Retrieve the employee's EmpID and the corresponding salary for the minimum z-score
min_z_score_employee <- hrdata_df[min_z_score_index, c("EmpID", "Salary")]

# Print the result
print(min_z_score_employee)
```

```
##     EmpID Salary
## 271 10271  45046
```

Interpretation

- The shape of the standardized salary distribution will be the same as the original salary distribution.
- The scales are different as the z-scores are standardized to have a mean of 0 and a standard deviation of 1.

3. Compute descriptive statistics for the standardized Salary variable. Report your results and produce a frequency distribution for the standardized Salary scores. Compare this distribution to the one you produced in Question 1. Are they the same or different? Explain using both your graphical results and words.

```r
# Monthly salary summary statistics
summary(hrdata_df$Monthly_Salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3754    4625    5234    5752    6003   20833
```

```r
# Mean
mean_monthlysalary <- mean(hrdata_df$Monthly_Salary, na.rm = TRUE)

# Median
median_Monthly_Salary <- median(hrdata_df$Monthly_Salary, na.rm = TRUE)

# Mode
mode_monthlysalary <- function(x) {
  uniq_x <- unique(x)
  uniq_x[which.max(tabulate(match(x, uniq_x)))]
}
mode_monthly_salary_value <- mode_monthlysalary(hrdata_df$Monthly_Salary)

print(paste("The mean monthly salary:", format(round(mean_monthlysalary, 2), nsmall = 2)))
```

```
## [1] "The mean monthly salary: 5751.72"
```

```r
print(paste("The median monthly salary is:", median_Monthly_Salary))
```

```
## [1] "The median monthly salary is: 5234.17"
```

```r
print(paste("The mode monthly salary is:", mode_monthly_salary_value))
```

```
## [1] "The mode monthly salary is: 4817.92"
```

Interpretation

- The monthly salary distribution retains the same overall shape as the original salary reported in Question 1 (right-skewed) but the scale is adjusted to z-scores.
- In the original distribution, the mean salary was around $69,020.68 with a standard deviation of $25,156.64. After standardization, the distribution is centered around 0.

4. Compute descriptive statistics for the standardized Salary variable. Report your results and produce a frequency distribution for the standardized Salary scores. Compare this distribution to the one you produced in Question 1. Are they the same or different? Explain using both your graphical results and words.

```r
# Monthly salary summary statistics
summary(hrdata_df$Monthly_Salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3754    4625    5234    5752    6003   20833
```

```r
# Mean
mean_monthlysalary <- mean(hrdata_df$Monthly_Salary, na.rm = TRUE)

# Median
median_Monthly_Salary <- median(hrdata_df$Monthly_Salary, na.rm = TRUE)

# Mode
mode_monthlysalary <- function(x) {
  uniq_x <- unique(x)
  uniq_x[which.max(tabulate(match(x, uniq_x)))]
}
mode_monthly_salary_value <- mode_monthlysalary(hrdata_df$Monthly_Salary)

print(paste("Mean monthly salary:", format(round(mean_monthlysalary, 2), nsmall = 2)))
```

```
## [1] "Mean monthly salary: 5751.72"
```

```r
print(paste("The median monthly salary is:", median_Monthly_Salary))
```

```
## [1] "The median monthly salary is: 5234.17"
```

```r
print(paste("The mode monthly salary is:", mode_monthly_salary_value))
```

```
## [1] "The mode monthly salary is: 4817.92"
```

Interpreation

- The overall distribution remains positively skewed, as was the case with the original salary distribution in Question 1.

5. Compute descriptive statistics for the monthly salary variable you created at the beginning of this project. Report your results and produce a frequency distribution for these scores. Compare this distribution to the ones you produced in Questions 1 & 3. Are they the same or different? Explain using both your graphical results and words

```r
# Monthly salary summary statistics
summary(hrdata_df$Monthly_Salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3754    4625    5234    5752    6003   20833
```

```r
# Mean
mean_monthlysalary <- mean(hrdata_df$Monthly_Salary, na.rm = TRUE)

# Median
median_Monthly_Salary <- median(hrdata_df$Monthly_Salary, na.rm = TRUE)

# Mode
mode_monthlysalary <- function(x) {
  uniq_x <- unique(x)
  uniq_x[which.max(tabulate(match(x, uniq_x)))]
}
mode_monthly_salary_value <- mode_monthlysalary(hrdata_df$Monthly_Salary)

print(paste("Mean monthly salary:", format(round(mean_monthlysalary, 2), nsmall = 2)))
```

```
## [1] "Mean monthly salary: 5751.72"
```

```r
print(paste("The median monthly salary is:", median_Monthly_Salary))
```

```
## [1] "The median monthly salary is: 5234.17"
```

```r
print(paste("The mode monthly salary is:", mode_monthly_salary_value))
```

```
## [1] "The mode monthly salary is: 4817.92"
```

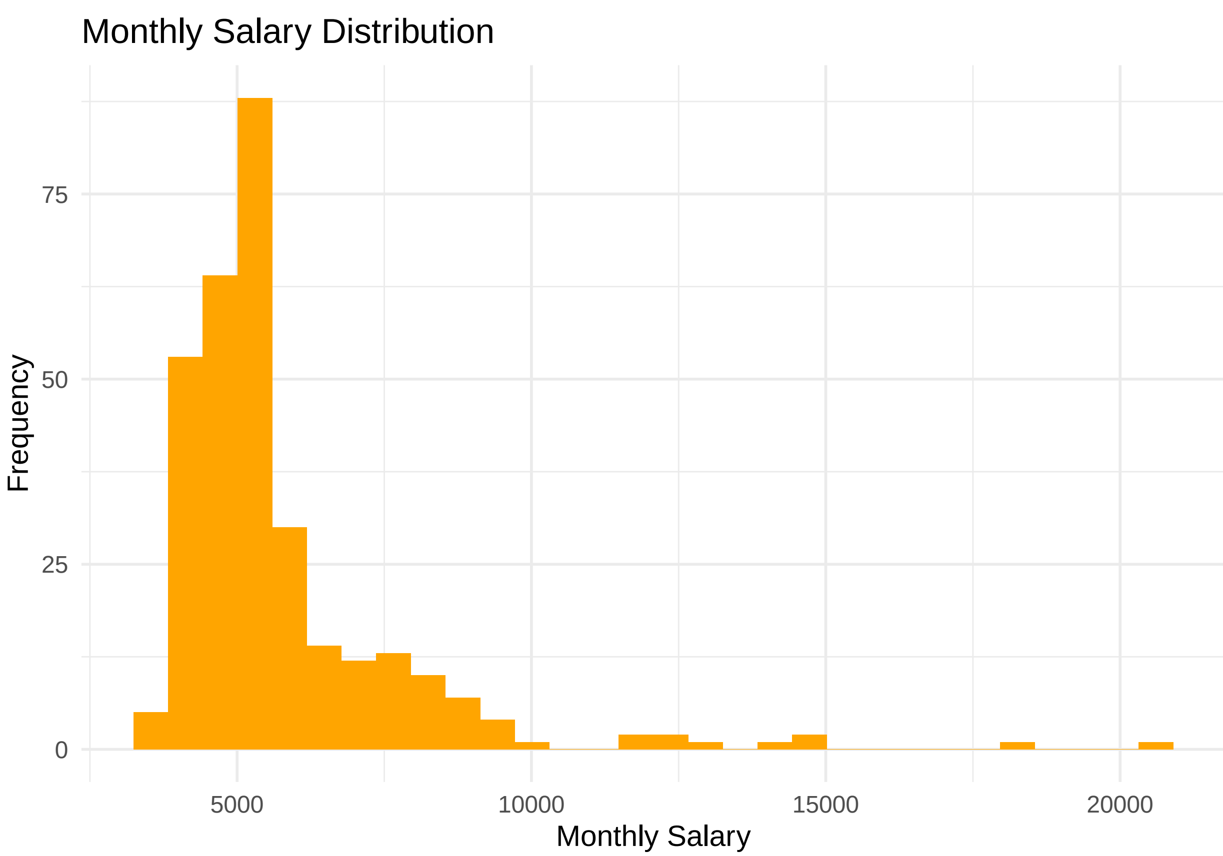
```r
# Frequency table
monthly_salary_table <- table(hrdata_df$Monthly_Salary)
print(monthly_salary_table)
```

```
##
## 3753.83 3755.75 3759.58 3782.92 3786.08 3833.17 3843.33 3861.25
##       1       1       1       1       1       1       1       1
##    3869 3869.17 3887.83 3888.67 3894.83 3899.92 3903.08  3916.5
##       1       1       1       1       1       1       1       1
## 3916.75 3934.25 3951.17 3952.83 3979.17 3986.42 3996.75 4023.75
##       1       1       1       1       1       1       1       1
## 4034.42 4041.25 4042.75    4074 4104.67 4147.75    4160  4181.5
##       1       1       1       1       1       1       1       1
##  4189.5 4197.75 4202.33 4205.83 4206.83 4229.17 4235.42 4243.58
##       1       1       1       1       1       1       1       1
## 4253.67 4271.58 4278.08 4292.08 4314.75 4325.67 4326.67 4338.08
##       1       1       1       1       1       1       1       1
## 4340.58 4348.08 4354.08 4375.42 4383.25 4385.33  4389.5    4399
##       1       1       1       1       1       1       1       1
```

```
## 4403.83 4415.33 4418.17 4421.67 4430.92 4431.67 4432.42  4437.5
##       1       1       1       1       1       1       1       1
## 4447.17 4457.67 4463.67 4500.42    4511 4519.75 4523.75 4555.83
##       1       1       1       1       1       1       1       1
##    4569 4577.75 4583.33    4595 4609.58 4618.75  4631.5 4640.67
##       1       1       1       1       1       1       1       1
##  4643.5    4650 4656.25 4663.75 4678.92 4679.08 4691.17 4694.92
##       1       1       1       1       1       1       1       1
## 4737.25 4749.25 4797.33 4797.92 4798.58 4812.33 4817.92  4819.5
##       1       1       1       1       1       1       2       1
## 4821.58  4829.5 4831.25  4838.5 4850.58 4856.08 4856.25 4864.17
##       1       1       1       1       1       1       1       1
## 4864.25 4876.92  4877.5 4892.42 4911.58 4918.83    4927 4928.67
##       1       1       1       1       1       1       1       1
## 4935.92  4936.5 4947.08 4947.42  4947.5    4956 4977.33    4991
##       1       1       1       1       1       1       1       1
## 5005.83    5010  5022.5 5028.33 5031.67 5036.33 5037.17 5052.25
##       1       1       1       1       1       1       1       1
## 5054.67 5060.33 5062.83 5096.17  5103.5 5112.42 5112.92  5118.5
##       1       1       1       1       2       1       1       1
## 5129.58 5130.67    5132    5138 5144.08 5150.75 5153.67  5163.5
##       1       1       1       1       1       1       1       1
## 5171.75 5172.08 5172.33 5180.17 5198.75 5202.08 5208.83  5209.5
##       1       1       1       1       1       1       1       1
## 5221.58 5234.17  5242.5 5246.42    5250 5250.25 5252.08 5254.25
##       1       1       1       1       1       1       2       1
##    5259 5274.25 5276.83 5279.42 5281.75 5285.83  5287.5 5289.83
##       1       1       1       1       1       1       1       1
## 5292.92 5306.33 5306.83 5307.92 5313.58 5317.75 5323.17 5331.08
##       1       1       1       1       1       1       1       1
## 5335.08 5338.08 5338.83 5353.83 5364.58 5366.42 5376.67 5393.67
##       1       1       1       1       1       1       1       1
## 5394.83 5398.83 5401.33 5409.92 5412.92 5414.25 5415.92 5416.25
##       1       1       1       1       1       1       1       1
## 5440.67  5442.5 5475.58 5476.17 5477.42 5491.08 5491.83 5506.17
##       1       1       1       1       1       1       1       1
## 5512.42 5536.75 5545.08 5549.42  5561.5 5567.33 5568.75    5598
##       1       1       1       1       1       1       1       1
## 5603.08 5604.25 5670.92 5674.92 5681.83 5700.58 5723.17 5735.75
##       1       1       1       1       1       1       1       1
##  5741.5 5749.92 5778.33 5844.25 5848.92 5872.33 5875.58 5878.75
##       1       1       1       1       1       1       1       1
## 5885.08 5944.92 5975.58 5981.33 5988.33 5997.17 6008.83 6016.83
##       1       1       1       1       1       1       1       1
## 6038.33 6050.75 6053.33 6082.67 6110.83  6185.5 6186.75 6192.67
##       1       1       1       1       1       1       1       1
## 6193.83 6201.42 6222.42 6223.25 6234.42 6265.67 6273.42 6335.75
##       1       1       1       1       1       1       1       1
## 6474.33 6492.92 6709.33 6798.67  6896.5  6923.5 6946.92 6962.67
##       1       1       1       1       1       1       1       1
## 6972.25 7075.25 7085.67  7184.5 7297.08 7318.83 7326.75 7377.25
##       1       1       1       1       1       1       1       1
## 7414.67    7441 7490.25 7508.33    7694 7694.08 7749.08 7753.83
##       1       1       1       1       1       1       1       1
```

```
##  7757.75  7767.17     7783  7796.17  7971.67  7993.33  8068.33  8166.58
##        1        1        1        1        1        1        1        1
##  8251.67  8273.33  8279.25  8335.92     8368  8433.25  8634.42  8703.08
##        1        1        1        1        1        1        1        1
##  8807.33  8808.33  8863.92   8935.5  9082.25  9166.67  9244.08  9499.92
##        1        1        1        1        1        1        1        1
##  9566.67    10000    11574 11743.33 12416.58 12524.17 13083.33 14208.33
##        1        1        1        1        1        1        1        1
## 14833.33    15000 18370.83 20833.33
##        1        1        1        1
```

```r
# Plot the monthly salary distribution
hrdata_df %>%
  ggplot(aes(x = Monthly_Salary)) +
  geom_histogram(fill = "orange", bins = 30) +
  labs(title = "Monthly Salary Distribution", x = "Monthly Salary", y = "Frequency") +
  theme_minimal()
```



Interpreation

- The monthly salary distribution retains the same overall shape as the original salary reported in both Questions 1 and 3 (right-skewed). The data is just scaled down by 12 as this is the number of months in a year.
- The overall distribution remains positively skewed, as was the case with the original salary distributions in Questions 1 and 3.

Compute simple descriptive statistics for the PerfSocreID variable. In addition, produce a visualization of the frequency distribution of PerfScoreID.

6. Report the descriptive statistics along with the frequency distribution and provide a detailed interpretation of how you would characterize this variable.

```
### Summary statistics
summary(hrdata_df$PerfScoreID)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   2.977   3.000   4.000
```

```
### Mean
mean_perfscoreid <- mean(hrdata_df$PerfScoreID, na.rm = TRUE)
print(mean_perfscoreid)
```

```
## [1] 2.977492
```

```
#### Median
median_perfscoreid <- median(hrdata_df$PerfScoreID, na.rm = TRUE)
print(median_perfscoreid)
```

```
## [1] 3
```

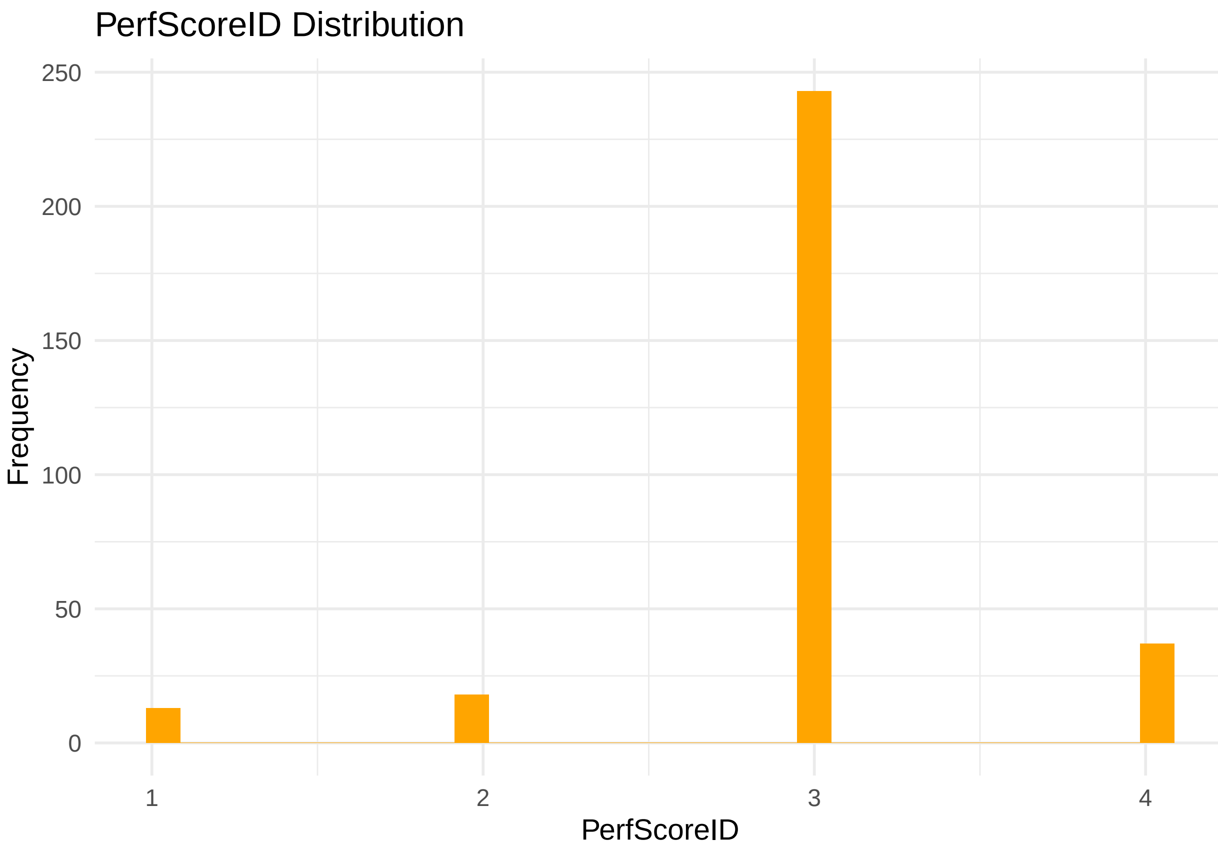
```
#### Mode
mode_perfscoreid <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
mode_perfscoreid_value <- mode_perfscoreid(hrdata_df$PerfScoreID)
print(mode_perfscoreid_value)
```

```
## [1] 3
```

```
### Frequency table
perfscoreid_table <- table(hrdata_df$PerfScoreID)
print(perfscoreid_table)
```

```
##
##   1   2   3   4
##  13  18 243  37
```

```
#### Plot
hrdata_df %>%
  ggplot(aes(x = PerfScoreID)) +
  geom_histogram(fill = "orange", bins = 30) +
  labs(title = "PerfScoreID Distribution", x = "PerfScoreID", y = "Frequency") +
  theme_minimal()
```



Interpretation

```r
print(paste("The mean PerfScoreID is", mean_perfscoreid))
```

```
## [1] "The mean PerfScoreID is 2.97749196141479"
```

```r
print(paste("The median PerfScoreID is", median_perfscoreid))
```

```
## [1] "The median PerfScoreID is 3"
```

```r
print(paste("The mode PerfScoreID is", mode_perfscoreid_value))
```

```
## [1] "The mode PerfScoreID is 3"
```

```r
print(paste("The frequency distribution of salaries is shown in the bar plot above."))
```

```
## [1] "The frequency distribution of salaries is shown in the bar plot above."
```

- The PerfScoreID distribution is strongly right-skewed. Most employees (243 out of 311) received a score of 3, indicating that 3 is the most common (modal) performance score.
- The mean is very close to the median (both near 3), further highlighting that the majority of employees are clustered around a performance score of 3.

7. Which employee (ID number) has the largest z-score on PerfScoreID and what is the z-score for this person? Which employee (ID number) has the smallest z- score on PerfScoreID and what is the z-score for this person?

```r
# Calculate the z-scores for the PerfScoreID column
hrdata_df <- hrdata_df %>%
  mutate(perfscoreid_zscore = scale(PerfScoreID))

# Print the z-scores
print(hrdata_df$perfscoreid_zscore)
```

```
##              [,1]
##   [1,] 1.74170930
##   [2,] 1.74170930
##   [3,] 1.74170930
##   [4,] 1.74170930
##   [5,] 1.74170930
##   [6,] 1.74170930
##   [7,] 1.74170930
##   [8,] 1.74170930
##   [9,] 1.74170930
##  [10,] 1.74170930
##  [11,] 1.74170930
##  [12,] 1.74170930
##  [13,] 1.74170930
##  [14,] 1.74170930
##  [15,] 1.74170930
##  [16,] 1.74170930
##  [17,] 1.74170930
##  [18,] 1.74170930
##  [19,] 1.74170930
##  [20,] 1.74170930
##  [21,] 1.74170930
##  [22,] 1.74170930
##  [23,] 1.74170930
##  [24,] 1.74170930
##  [25,] 1.74170930
##  [26,] 1.74170930
##  [27,] 1.74170930
##  [28,] 1.74170930
##  [29,] 1.74170930
##  [30,] 1.74170930
##  [31,] 1.74170930
##  [32,] 1.74170930
##  [33,] 1.74170930
##  [34,] 1.74170930
##  [35,] 1.74170930
##  [36,] 1.74170930
##  [37,] 1.74170930
##  [38,] 0.03833951
```

```
## [39,] 0.03833951
## [40,] 0.03833951
## [41,] 0.03833951
## [42,] 0.03833951
## [43,] 0.03833951
## [44,] 0.03833951
## [45,] 0.03833951
## [46,] 0.03833951
## [47,] 0.03833951
## [48,] 0.03833951
## [49,] 0.03833951
## [50,] 0.03833951
## [51,] 0.03833951
## [52,] 0.03833951
## [53,] 0.03833951
## [54,] 0.03833951
## [55,] 0.03833951
## [56,] 0.03833951
## [57,] 0.03833951
## [58,] 0.03833951
## [59,] 0.03833951
## [60,] 0.03833951
## [61,] 0.03833951
## [62,] 0.03833951
## [63,] 0.03833951
## [64,] 0.03833951
## [65,] 0.03833951
## [66,] 0.03833951
## [67,] 0.03833951
## [68,] 0.03833951
## [69,] 0.03833951
## [70,] 0.03833951
## [71,] 0.03833951
## [72,] 0.03833951
## [73,] 0.03833951
## [74,] 0.03833951
## [75,] 0.03833951
## [76,] 0.03833951
## [77,] 0.03833951
## [78,] 0.03833951
## [79,] 0.03833951
## [80,] 0.03833951
## [81,] 0.03833951
## [82,] 0.03833951
## [83,] 0.03833951
## [84,] 0.03833951
## [85,] 0.03833951
## [86,] 0.03833951
## [87,] 0.03833951
## [88,] 0.03833951
## [89,] 0.03833951
## [90,] 0.03833951
## [91,] 0.03833951
## [92,] 0.03833951
```

```
##  [93,]  0.03833951
##  [94,]  0.03833951
##  [95,]  0.03833951
##  [96,]  0.03833951
##  [97,]  0.03833951
##  [98,]  0.03833951
##  [99,]  0.03833951
## [100,]  0.03833951
## [101,]  0.03833951
## [102,]  0.03833951
## [103,]  0.03833951
## [104,]  0.03833951
## [105,]  0.03833951
## [106,]  0.03833951
## [107,]  0.03833951
## [108,]  0.03833951
## [109,]  0.03833951
## [110,]  0.03833951
## [111,]  0.03833951
## [112,]  0.03833951
## [113,]  0.03833951
## [114,]  0.03833951
## [115,]  0.03833951
## [116,]  0.03833951
## [117,]  0.03833951
## [118,]  0.03833951
## [119,]  0.03833951
## [120,]  0.03833951
## [121,]  0.03833951
## [122,]  0.03833951
## [123,]  0.03833951
## [124,]  0.03833951
## [125,]  0.03833951
## [126,]  0.03833951
## [127,]  0.03833951
## [128,]  0.03833951
## [129,]  0.03833951
## [130,]  0.03833951
## [131,]  0.03833951
## [132,]  0.03833951
## [133,]  0.03833951
## [134,]  0.03833951
## [135,]  0.03833951
## [136,]  0.03833951
## [137,]  0.03833951
## [138,]  0.03833951
## [139,]  0.03833951
## [140,]  0.03833951
## [141,]  0.03833951
## [142,]  0.03833951
## [143,]  0.03833951
## [144,]  0.03833951
## [145,]  0.03833951
## [146,]  0.03833951
```

```
## [147,]  0.03833951
## [148,]  0.03833951
## [149,]  0.03833951
## [150,]  0.03833951
## [151,]  0.03833951
## [152,]  0.03833951
## [153,]  0.03833951
## [154,]  0.03833951
## [155,]  0.03833951
## [156,]  0.03833951
## [157,]  0.03833951
## [158,]  0.03833951
## [159,]  0.03833951
## [160,]  0.03833951
## [161,]  0.03833951
## [162,]  0.03833951
## [163,]  0.03833951
## [164,]  0.03833951
## [165,]  0.03833951
## [166,]  0.03833951
## [167,]  0.03833951
## [168,]  0.03833951
## [169,]  0.03833951
## [170,]  0.03833951
## [171,]  0.03833951
## [172,]  0.03833951
## [173,]  0.03833951
## [174,]  0.03833951
## [175,]  0.03833951
## [176,]  0.03833951
## [177,]  0.03833951
## [178,]  0.03833951
## [179,]  0.03833951
## [180,]  0.03833951
## [181,]  0.03833951
## [182,]  0.03833951
## [183,]  0.03833951
## [184,]  0.03833951
## [185,]  0.03833951
## [186,]  0.03833951
## [187,]  0.03833951
## [188,]  0.03833951
## [189,]  0.03833951
## [190,]  0.03833951
## [191,]  0.03833951
## [192,]  0.03833951
## [193,]  0.03833951
## [194,]  0.03833951
## [195,]  0.03833951
## [196,]  0.03833951
## [197,]  0.03833951
## [198,]  0.03833951
## [199,]  0.03833951
## [200,]  0.03833951
```

```
## [201,]  0.03833951
## [202,]  0.03833951
## [203,]  0.03833951
## [204,]  0.03833951
## [205,]  0.03833951
## [206,]  0.03833951
## [207,]  0.03833951
## [208,]  0.03833951
## [209,]  0.03833951
## [210,]  0.03833951
## [211,]  0.03833951
## [212,]  0.03833951
## [213,]  0.03833951
## [214,]  0.03833951
## [215,]  0.03833951
## [216,]  0.03833951
## [217,]  0.03833951
## [218,]  0.03833951
## [219,]  0.03833951
## [220,]  0.03833951
## [221,]  0.03833951
## [222,]  0.03833951
## [223,]  0.03833951
## [224,]  0.03833951
## [225,]  0.03833951
## [226,]  0.03833951
## [227,]  0.03833951
## [228,]  0.03833951
## [229,]  0.03833951
## [230,]  0.03833951
## [231,]  0.03833951
## [232,]  0.03833951
## [233,]  0.03833951
## [234,]  0.03833951
## [235,]  0.03833951
## [236,]  0.03833951
## [237,]  0.03833951
## [238,]  0.03833951
## [239,]  0.03833951
## [240,]  0.03833951
## [241,]  0.03833951
## [242,]  0.03833951
## [243,]  0.03833951
## [244,]  0.03833951
## [245,]  0.03833951
## [246,]  0.03833951
## [247,]  0.03833951
## [248,]  0.03833951
## [249,]  0.03833951
## [250,]  0.03833951
## [251,]  0.03833951
## [252,]  0.03833951
## [253,]  0.03833951
## [254,]  0.03833951
```

```
## [255,]  0.03833951
## [256,]  0.03833951
## [257,]  0.03833951
## [258,]  0.03833951
## [259,]  0.03833951
## [260,]  0.03833951
## [261,]  0.03833951
## [262,]  0.03833951
## [263,]  0.03833951
## [264,]  0.03833951
## [265,]  0.03833951
## [266,]  0.03833951
## [267,]  0.03833951
## [268,]  0.03833951
## [269,]  0.03833951
## [270,]  0.03833951
## [271,]  0.03833951
## [272,]  0.03833951
## [273,]  0.03833951
## [274,]  0.03833951
## [275,]  0.03833951
## [276,]  0.03833951
## [277,]  0.03833951
## [278,]  0.03833951
## [279,]  0.03833951
## [280,] -1.66503028
## [281,] -1.66503028
## [282,] -1.66503028
## [283,] -1.66503028
## [284,] -1.66503028
## [285,] -1.66503028
## [286,] -1.66503028
## [287,] -1.66503028
## [288,] -1.66503028
## [289,] -1.66503028
## [290,] -1.66503028
## [291,] -1.66503028
## [292,] -1.66503028
## [293,] -1.66503028
## [294,] -1.66503028
## [295,] -1.66503028
## [296,] -1.66503028
## [297,] -1.66503028
## [298,] -3.36840007
## [299,] -3.36840007
## [300,] -3.36840007
## [301,] -3.36840007
## [302,] -3.36840007
## [303,] -3.36840007
## [304,] -3.36840007
## [305,]  0.03833951
## [306,] -3.36840007
## [307,] -3.36840007
## [308,] -3.36840007
```

```
## [309,] -3.36840007
## [310,] -3.36840007
## [311,] -3.36840007
## attr(,"scaled:center")
## [1] 2.977492
## attr(,"scaled:scale")
## [1] 0.5870716
```

```r
# Find the index of the employee with the largest z-score
max_z_score_index <- which.max(hrdata_df$perfscoreid_zscore)
print(max_z_score_index)
```

```
## [1] 1
```

```r
# Retrieve the employee's EmpID and the corresponding z-score
max_z_score_employee <- hrdata_df[max_z_score_index, c("EmpID", "PerfScoreID")]

# Print the result
print(max_z_score_employee)
```

```
##   EmpID PerfScoreID
## 1 10001           4
```

```r
# Find the index of the employee with the smallest z-score
min_z_score_index <- which.min(hrdata_df$perfscoreid_zscore)
print(min_z_score_index)
```

```
## [1] 298
```

```r
# Retrieve the employee's EmpID and the corresponding PerfScoreID for the minimum z-score
min_z_score_employee <- hrdata_df[min_z_score_index, c("EmpID", "PerfScoreID")]

# Print the result
print(min_z_score_employee)
```

```
##     EmpID PerfScoreID
## 298 10298           1
```

Interpretation

- The employee with the highest performance score (4) has a z-score of 1.7417, indicating that this employee's score is significantly above the mean (by about 1.74 standard deviations).
- The employee with the lowest performance score (1) has a z-score of -3.3684, indicating that their score is far below the mean.

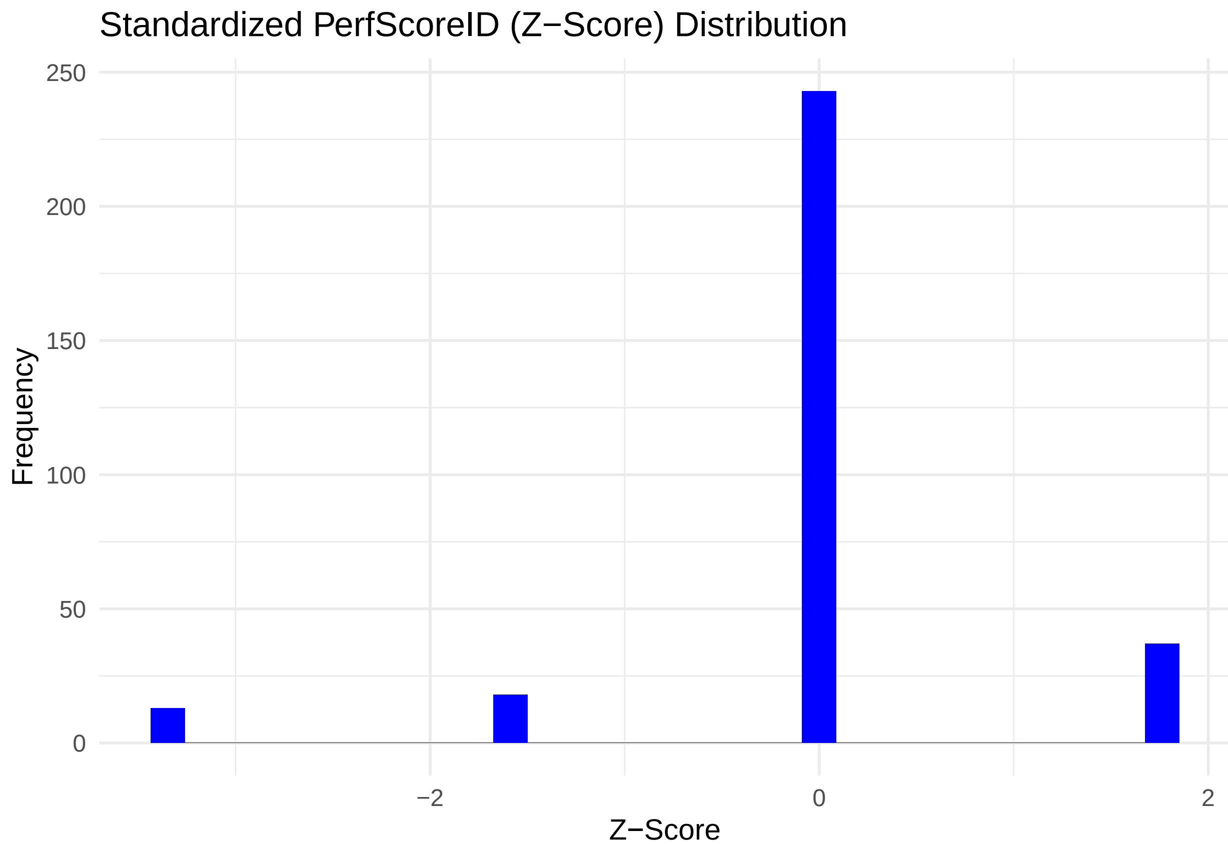
8. Compute descriptive statistics for the standardized PerfScoreID variable. Report your results and produce a frequency distribution for the standardized PerfScoreID scores. Compare this distribution to the one you produced in Question 6. Are they the same or different? Explain using both your graphical results and words.

```
# Calculate the z-scores for the PerfScoreID column
hrdata_df <- hrdata_df %>%
  mutate(perfscoreid_zscore = scale(PerfScoreID))

# Summary statistics
summary(hrdata_df$perfscoreid_zscore)

# Mean
mean_perfscoreid_zscore <- mean(hrdata_df$perfscoreid_zscore, na.rm = TRUE)
print(paste("Mean of PerfScoreID z-score:", mean_perfscoreid_zscore))

# Median
median_perfscoreid_zscore <- median(hrdata_df$perfscoreid_zscore, na.rm = TRUE)
print(paste("Median of PerfScoreID z-score:", median_perfscoreid_zscore))

# Mode
mode_perfscoreid_zscore <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
mode_perfscoreid_zscore_value <- mode_perfscoreid_zscore(hrdata_df$perfscoreid_zscore)
print(paste("Mode of PerfScoreID z-score:", mode_perfscoreid_zscore_value))
```

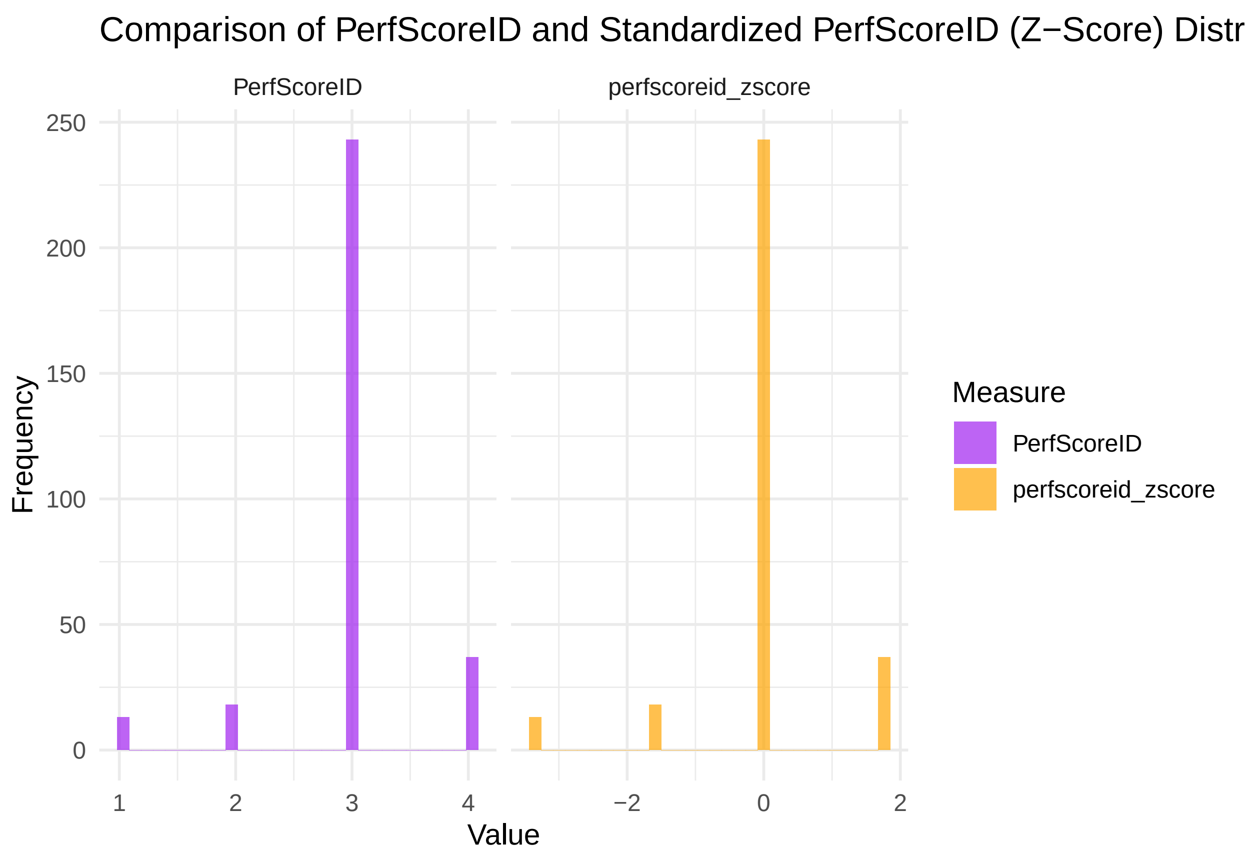
```
# Frequency table
perfscoreid_zscore_table <- table(hrdata_df$perfscoreid_zscore)
print(perfscoreid_zscore_table)
```

```
##
##   -3.3684000667055  -1.66503027687556 0.0383395129543715   1.74170930278431
##                 13                 18                243                 37
```

```
# Plot the PerfScoreID z-score
hrdata_df %>%
  ggplot(aes(x = perfscoreid_zscore)) +
  geom_histogram(fill = "blue", bins = 30) +
  labs(title = "Standardized PerfScoreID (Z-Score) Distribution", x = "Z-Score", y = "Frequency") +
  theme_minimal()
```




```r
# Plot of the PerfScoreID z-score and PerfScoreID
hrdata_df %>%
  pivot_longer(cols = c(PerfScoreID, perfscoreid_zscore),
               names_to = "Measure",
               values_to = "Value") %>%
  ggplot(aes(x = Value, fill = Measure)) +
  geom_histogram(bins = 30, alpha = 0.7) +
  facet_wrap(~ Measure, scales = "free_x") +   # Creates separate plots for PerfScoreID and Z-Score
  labs(title = "Comparison of PerfScoreID and Standardized PerfScoreID (Z-Score) Distributions",
       x = "Value", y = "Frequency") +
  theme_minimal() +
  scale_fill_manual(values = c("purple", "orange"))
```



Interpretation

- The z-score distribution shows that the bulk of employees' performance scores are close to the mean, with only a few employees having very high or very low performance scores.
- The shape of the distribution remains the same as the original PerfScoreID distribution in Question 6, but it is now centered around 0 with a standardized spread.

9. Square the PerfScoreID variable. Compute descriptive statistics for this new variable. Report your results and produce a frequency distribution for the squared scores. Compare this distribution to the ones you produced in Questions 1 & 3. Are they the same or different? Explain using both your graphical results and words.

```r
# Square of the PerfScoreID variable
squared_perfscoreid <- hrdata_df$PerfScoreID^2

# Summary statistics
summary(squared_perfscoreid)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   9.000   9.000   9.209   9.000  16.000
```

```r
# Mean
mean_squared_perfscoreid <- mean(squared_perfscoreid, na.rm = TRUE)

# Median
median_squared_perfscoreid <- median(squared_perfscoreid, na.rm = TRUE)

# Mode
mode_squared_perfscoreid <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
mode_squared_perfscoreid_value <- mode_squared_perfscoreid(squared_perfscoreid)

print(paste("Mean squared PerfScoreID:", format(round(mean_squared_perfscoreid, 2), nsmall = 2)))
```

```
## [1] "Mean squared PerfScoreID: 9.21"
```

```r
print(paste("Median squared PerfScoreID:", median_squared_perfscoreid))
```

```
## [1] "Median squared PerfScoreID: 9"
```

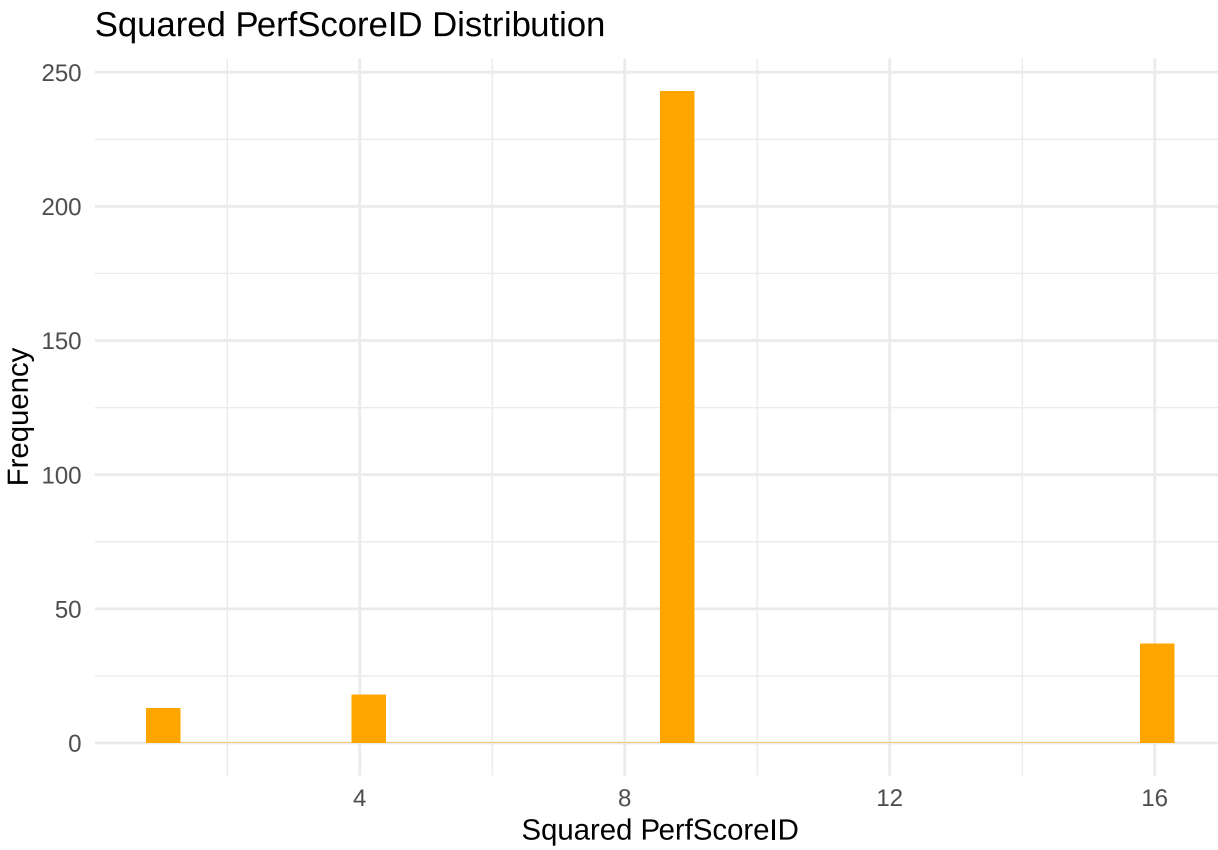
```r
print(paste("Mode squared PerfScoreID:", mode_squared_perfscoreid_value))
```

```
## [1] "Mode squared PerfScoreID: 9"
```

```r
# Frequency table
squared_perfscoreid_table <- table(squared_perfscoreid)
print(squared_perfscoreid_table)
```

```
## squared_perfscoreid
##   1   4   9  16
##  13  18 243  37
```

```r
# Plot the squared PerfScoreID
hrdata_df %>%
  mutate(squared_perfscoreid = PerfScoreID^2) %>%
  ggplot(aes(x = squared_perfscoreid)) +
  geom_histogram(fill = "orange", bins = 30) +
  labs(title = "Squared PerfScoreID Distribution", x = "Squared PerfScoreID", y = "Frequency") +
  theme_minimal()
```



Interpretation

- The squared transformation increases differences in scores, especially for employees with extreme performance scores.
- The median and mode remain 9, as most employees still have a performance score of 3 (which squares to 9).
- The shape of the distribution is different from the results from Questions 1 and 3 as the squared values accentuate the differences between scores resulting in a more spread-out distribution although still unimodal.