Benford

2024-12-14

Load the data

```
Benford1906 <- readRDS("Benford1906.RDS")
Benford1913 <- readRDS("Benford1913.RDS")
```

```
Benford1906
```

```
##                                Office Posted_Letters Received_Letters
## 1                          Abercrombie           5337             6795
## 2                             Aberdeen          84822            93309
## 3                            Aberfoyle              0                0
## 4                             Abermain          12096             3377
## 5                             Abington           3194             5175
## 6                         Acacia Creek           8650            18983
## 7                            Adaminiby          51389            55531
## 8                            Adamstown          56303            85659
## 9                              Adelong          88844           139673
## 10                 Adelong Crossing Place          44721            41366
## 11                         Adjungbilly           3490             4256
## 12                         Agnes Banks           4153             7576
## 13                               Airly          16235            20897
## 14                         Albion Park         123159           139311
## 15                               Albury         749204           657177
## 16                       Albury Railway              0                0
## 17                             Alectown           9289            19431
## 18                           Alexandria         109651           387583
## 19                               Alison           2529             6078
## 20                            Allandale           7428             9419
## 21                           Allynbrook          11623            14121
## 22                           Alstonville          63579            60754
## 23                               Amaroo           4178             6844
## 24                            Angledool              0                0
## 25                             Anna Bay           3214             6298
## 26                            Annandale         123100           476396
## 27                                Appin          12513            13056
## 28                              Arakoon              0                0
## 29                              Araluen          50861            72834
## 30                              Arcadia           5256            13929
## 31                             Ardgle'n           7698             7485
## 32                               Arding           1871             4822
## 33                         Argents Hill           2868             6879
## 34                               Argoon           3680             5761
## 35                          Ariah Park              0                0
## 36                             Arkstone           1360             2167
## 37                             Armatree           3323             4952
## 38                             Armidale         660267           633156
## 39                       Armidale Railway              0                0
## 40                              Arneliffe         254841           215074
## 41                          Arthurville           1156             2687
## 42                             Ashfield         218583           834178
## 43                              Ashford          18270            27252
## 44                           Ash Island           1285             2353
## 45                               Ashley           5329            17842
## 46                              Attunga          43276            53809
## 47                        Attunga Springs            772             3407
## 48                               Auburn          16746           196079
## 49                               Audley           3212             5519
## 50                             Austinmer           4581             3865
## 51                              Austral           3148             5632
## 52                        Australia Hotel              0                0
## 53                              Avisford           3489             5332
## 54                                 Avoca           5193             5598
## 55                                 Awaba           5742             5953
## 56                              Baan Baa          19997            20722
## 57                             Backwater           1498             2516
## 58                       Badgery's Creek           2955             3888
## 59                               Baerami           6846             7885
## 60                         Baker's Swamp           1957             2717
## 61                                Balala           2955             3648
```

```
## 4151               0             14        290    SA
## 4152               0              0        144    SA
## 4153               0              0         35    SA
## 4154               0              0        271    SA
## 4155               0              0        145    SA
## 4156               0              0         20    SA
## 4157               0              0         20    SA
## 4158               0              0         11    SA
## 4159               0              0         28    SA
## 4160               0              0         51    SA
## 4161               0              0        220    SA
## 4162               0              0        119    SA
## 4163               0              0         14    SA
## 4164             351             13        250    SA
## 4165               0              0        120    SA
## 4166               0              0         43    SA
##  [ reached 'max' / getOption("max.print") -- omitted 4659 rows ]
```

1. Clean the dataset

2. Descriptic statistics

```
colnames(Benford1906)
```

```
##  [1] "Office"            "Posted_Letters"    "Received_Letters"
##  [4] "Telegram_Number"   "SavingsDep_Number" "SavingsDep_Value"
##  [7] "Revenues_Postal"   "Revenues_Tele"     "Revenues_Money"
## [10] "Revenues_Total"    "Population"        "State"
```

```
colnames(Benford1913)
```

```
##  [1] "Office"               "Type"                 "Posted_Letters"
##  [4] "Received_Letters"     "Posted_Newspapers"    "Received_Newspapers"
##  [7] "Posted_Parcels"       "Received_Parcels"     "Telegram_Number"
## [10] "Telegram_Value"       "CallsOut_Number"      "CallsOut_Value"
## [13] "SavingsDep_Number"    "SavingsDep_Value"     "Revenues_Postal"
## [16] "Revenues_Telegraph"   "Revenues_Telephone"   "Revenues_MoneyOrder"
## [19] "Revenues_PostalNotes" "Revenues_Total"       "Pensions_Number"
## [22] "Pensions_Value"       "Population"           "State"
```

Demographics:

Population: Total population for each state or region. Postal Activity:

Office: Number of postal offices. Posted_Letters and Received_Letters Posted_Newspapers and Received_Newspapers Posted_Parcels and Received_Parcels: Number of parcels processed. Telecommunication:

Telegram_Number and Telegram_Value CallsOut_Number and CallsOut_Value Savings and Financial Services:

SavingsDep_Number and SavingsDep_Value: Number and total value of savings deposits. Revenues_Money, Revenues_MoneyOrder, Revenues_PostalNotes: Revenue generated from financial services.

```
summary(Benford1906)
```

```
##     Office          Posted_Letters    Received_Letters  Telegram_Number
##  Length:5810        Min.   :      0   Min.   :      0   Min.   :     0.0
##  Class :character   1st Qu.:   1919   1st Qu.:      0   1st Qu.:     0.0
##  Mode  :character   Median :   5042   Median :    898   Median :    80.0
##                     Mean   :  35328   Mean   :  24775   Mean   :  1533.8
##                     3rd Qu.:  17029   3rd Qu.:  10535   3rd Qu.:   778.5
##                     Max.   :8156244   Max.   :4343624   Max.   :328616.0
##  SavingsDep_Number SavingsDep_Value Revenues_Postal   Revenues_Tele
##  Min.   :    0.0   Min.   :    0    Min.   :    0.0   Min.   :    0
##  1st Qu.:    0.0   1st Qu.:    0    1st Qu.:   12.0   1st Qu.:    0
##  Median :    0.0   Median :    0    Median :   34.0   Median :    2
##  Mean   :  167.3   Mean   : 1174    Mean   :  233.2   Mean   :  105
##  3rd Qu.:    0.0   3rd Qu.:    0    3rd Qu.:  117.0   3rd Qu.:   34
##  Max.   :15387.0   Max.   :91237    Max.   :25692.0   Max.   :22692
##  Revenues_Money    Revenues_Total      Population        State
##  Min.   :   0.00   Min.   :    0.0   Min.   :     0   Length:5810
##  1st Qu.:   1.00   1st Qu.:   15.0   1st Qu.:    24   Class :character
##  Median :   2.00   Median :   41.0   Median :   115   Mode  :character
##  Mean   :  14.69   Mean   :  355.0   Mean   :   640
##  3rd Qu.:  10.00   3rd Qu.:  161.8   3rd Qu.:   300
##  Max.   :1073.00   Max.   :34696.0   Max.   :132468
```

```
summary(Benford1913)
```

```
##     Office              Type           Posted_Letters    Received_Letters
##  Length:8825        Length:8825        Min.   :      0   Min.   :      0
##  Class :character   Class :character   1st Qu.:   1515   1st Qu.:   2593
##  Mode  :character   Mode  :character   Median :   4109   Median :   6430
##                                        Mean   :  32614   Mean   :  39381
##                                        3rd Qu.:  13650   3rd Qu.:  18616
##                                        Max.   :7512551   Max.   :4618646
##  Posted_Newspapers Received_Newspapers Posted_Parcels    Received_Parcels
##  Min.   :      0   Min.   :     0      Min.   :    0.0   Min.   :    0.0
##  1st Qu.:     72   1st Qu.:  1300      1st Qu.:    0.0   1st Qu.:    0.0
##  Median :    286   Median :  3608      Median :   12.0   Median :   52.0
##  Mean   :   6807   Mean   : 12541      Mean   :  222.5   Mean   :  370.8
##  3rd Qu.:    969   3rd Qu.:  8983      3rd Qu.:   63.0   3rd Qu.:  226.0
##  Max.   :2315911   Max.   :950288      Max.   :60808.0   Max.   :26775.0
##  Telegram_Number  Telegram_Value     CallsOut_Number   CallsOut_Value
##  Min.   :     0   Min.   :    0.00   Min.   :    0.0   Min.   :   0.00
##  1st Qu.:     0   1st Qu.:    0.00   1st Qu.:    0.0   1st Qu.:   0.00
##  Median :    30   Median :    1.00   Median :    0.0   Median :   0.00
##  Mean   :  1228   Mean   :   67.22   Mean   :  617.1   Mean   :  14.83
##  3rd Qu.:   448   3rd Qu.:   19.00   3rd Qu.:  360.0   3rd Qu.:   6.00
##  Max.   :250504   Max.   :35494.00   Max.   :75117.0   Max.   :3543.00
##  SavingsDep_Number SavingsDep_Value  Revenues_Postal   Revenues_Telegraph
##  Min.   :   0.00   Min.   :    0.0   Min.   :    0.0   Min.   :    0.00
##  1st Qu.:   0.00   1st Qu.:    0.0   1st Qu.:    5.0   1st Qu.:    0.00
##  Median :   0.00   Median :    0.0   Median :   21.0   Median :    1.00
##  Mean   :  33.77   Mean   :  315.1   Mean   :  181.7   Mean   :   66.26
##  3rd Qu.:   0.00   3rd Qu.:    0.0   3rd Qu.:   73.0   3rd Qu.:   19.00
##  Max.   :4438.00   Max.   :29273.0   Max.   :32793.0   Max.   :35494.00
##  Revenues_Telephone Revenues_MoneyOrder Revenues_PostalNotes Revenues_Total
##  Min.   :    0.0    Min.   :  0.000     Min.   :  0.000      Min.   :    0
##  1st Qu.:    0.0    1st Qu.:  0.000     1st Qu.:  0.000      1st Qu.:    8
##  Median :    0.0    Median :  0.000     Median :  1.000      Median :   28
##  Mean   :   62.9    Mean   :  6.157     Mean   :  7.515      Mean   :  326
##  3rd Qu.:    9.0    3rd Qu.:  0.000     3rd Qu.:  5.000      3rd Qu.:  111
##  Max.   :24369.0    Max.   :719.000     Max.   :493.000      Max.   :43647
##  Pensions_Number   Pensions_Value      Population          State
##  Min.   :    0.0   Min.   :    0.0   Min.   :     0.0   Length:8825
##  1st Qu.:    0.0   1st Qu.:    0.0   1st Qu.:     0.0   Class :character
##  Median :    0.0   Median :    0.0   Median :    85.0   Mode  :character
##  Mean   :  273.9   Mean   :  248.7   Mean   :   588.2
##  3rd Qu.:   52.0   3rd Qu.:   26.0   3rd Qu.:   250.0
##  Max.   :45370.0   Max.   :44235.0   Max.   :100000.0
```

3. Use the Benford’s law to verify/demonstrate or do anything with the date?

Benford’s Law, also known as the “First-Digit Law,” is a probability distribution that predicts the frequency of the leading digits (1 through 9) in naturally occurring datasets. According to this law, lower digits (like 1) appear as the leading digit more frequently than higher digits (like 9).

Variables Suitable for Benford’s Law

Benford’s Law is applicable to datasets with wide ranges and values that grow exponentially:

- Posted_Letters, Received_Letters
- Telegram_Number, SavingsDep_Number, SavingsDep_Value
- Revenues_Postal, Revenues_Telegraph, Revenues_Total
- Population

#Apply Benford law

###Extract first digits

```
# Extract leading digits for 1906
leading_digits_1906 <- substr(as.character(Benford1906), 1, 1)
leading_digits_1906 <- as.numeric(leading_digits_1906)
```

```
## Warning: NAs introduced by coercion
```

```
# Extract leading digits for 1913
leading_digits_1913 <- substr(as.character(Benford1913), 1, 1)
leading_digits_1913 <- as.numeric(leading_digits_1913)
```

```
## Warning: NAs introduced by coercion
```

“NAs introduced by coercion” occurs when the as.numeric() function encounters non-numeric characters or empty strings. Benford1906 or Benford1913 data could contain missing values, special characters, or invalid entries.

```
# Remove non-numeric values and missing entries from Benford1906
#Benford1906_clean <- Benford1906[!is.na(as.numeric(as.character(Benford1906))),]

# Remove non-numeric values and missing entries from Benford1913
#Benford1913_clean <- Benford1913[!is.na(as.numeric(as.character(Benford1913)))]

 # For 1906
#leading_digits_1906 <- substr(as.character(Benford1906_clean), 1, 1)
#leading_digits_1906 <- as.numeric(leading_digits_1906)

# For 1913
#leading_digits_1913 <- substr(as.character(Benford1913_clean), 1, 1)
#leading_digits_1913 <- as.numeric(leading_digits_1913)
```

We must extract the leading digits by each variable

Posted letters

```
# Remove non-numeric values from Posted_letters
Benford1906_Posted_letters <- Benford1906$Posted_Letters
leading_digits_1906_Posted_letters<-as.numeric(substr(as.character(Benford1906_Posted_letters),1,1))
library(BenfordTests)
chisq.benftest(Benford1906_Posted_letters)
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Posted_letters
## chisq = 111.23, p-value < 2.2e-16
```

```
test<-chisq.benftest(Benford1906$Revenues_Tele[Benford1906$Revenues_Tele>0])
chisq.benftest(Benford1906_Posted_letters[Benford1906_Posted_letters>0],digits=2)
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Posted_letters[Benford1906_Posted_letters > 0]
## chisq = 114.45, p-value = 0.03594
```

```
chisq.benftest(Benford1906_Posted_letters[Benford1906_Posted_letters>0],digits=3)
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Posted_letters[Benford1906_Posted_letters > 0]
## chisq = 979.33, p-value = 0.0317
```

Benford law

Benford's distribution for the first digit is given by:

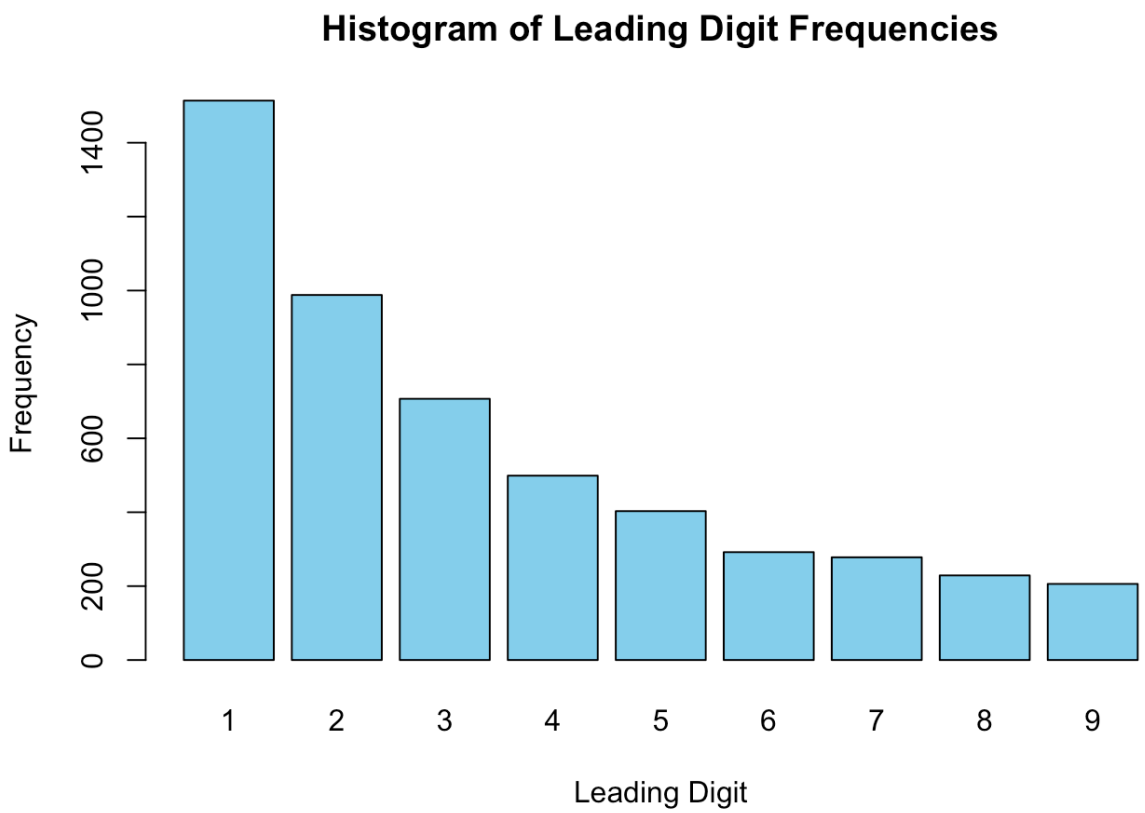
$P(d)$=log10(1+1/d) where d=1,2,…,9.

```
# Expected Benford probabilities
benford_probs <- log10(1 + 1 / (1:9))
```

Count the occurrences of each leading digit in the datasets:

try all possible colums to find good pvalue, draw histogram, compare the significant and non-significant ones.

```
leading_digits_1906_Posted_letters<-as.numeric(substr(as.character(Benford1906_Posted_letters),1,1))

#Calculate the frequency of each leading digit (1 to 9)
digit_frequencies_posted_letters <- table(factor(leading_digits_1906_Posted_letters, levels = 1:9))
barplot(
  digit_frequencies_posted_letters,
  main = "Histogram of Leading Digit Frequencies",
  xlab = "Leading Digit",
  ylab = "Frequency",
  col = "skyblue",
  names.arg = 1:9  # Explicitly set the labels for 1 to 9
)
```

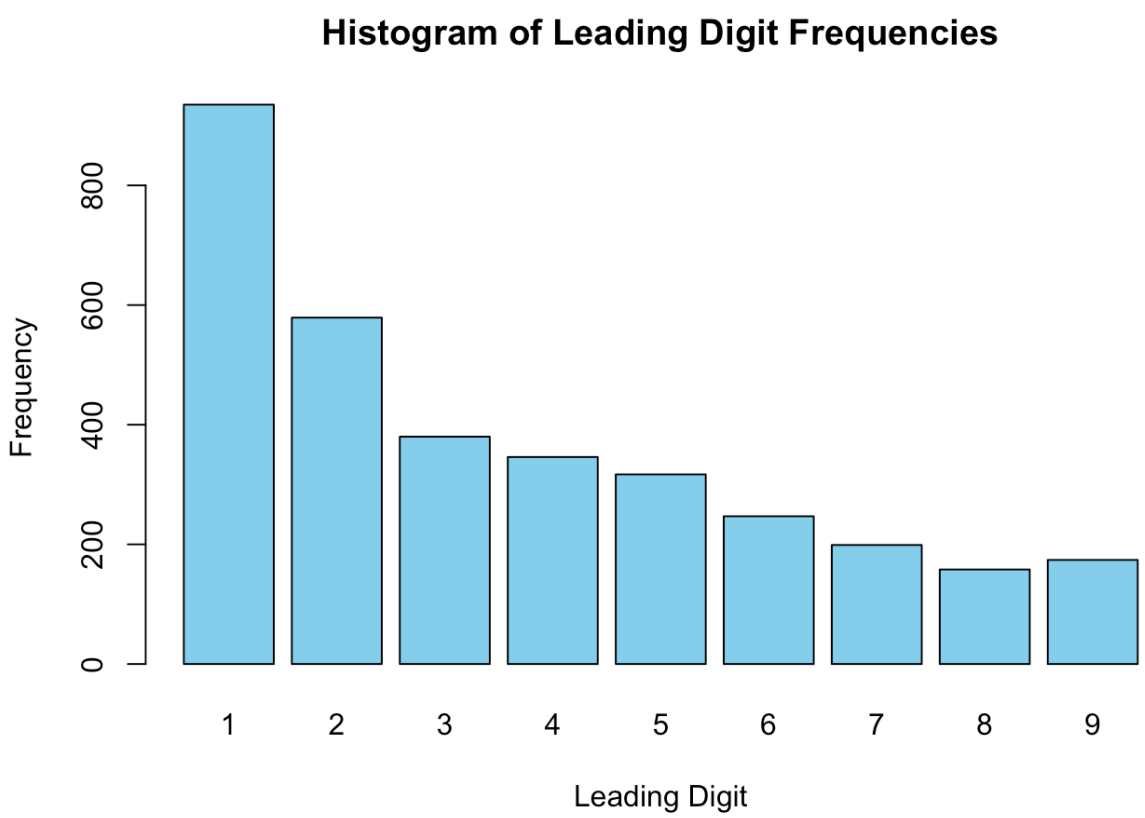


Received letters

```
Benford1906_Received_letters<-Benford1906$Received_Letters
leading_digits_1906_Received_letters<-as.numeric(substr(as.character(Benford1906_Received_letters),1,
1))
chisq.benftest(Benford1906_Received_letters[Benford1906_Received_letters>0])
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Received_letters[Benford1906_Received_letters > 0]
## chisq = 26.911, p-value = 0.0007325
```

```
#Calculate the frequency of each leading digit (1 to 9)
digit_frequencies_received_letters <- table(factor(leading_digits_1906_Received_letters, levels = 1:
9))
barplot(
  digit_frequencies_received_letters,
  main = "Histogram of Leading Digit Frequencies",
  xlab = "Leading Digit",
  ylab = "Frequency",
  col = "skyblue",
  names.arg = 1:9  # Explicitly set the labels for 1 to 9
)
```

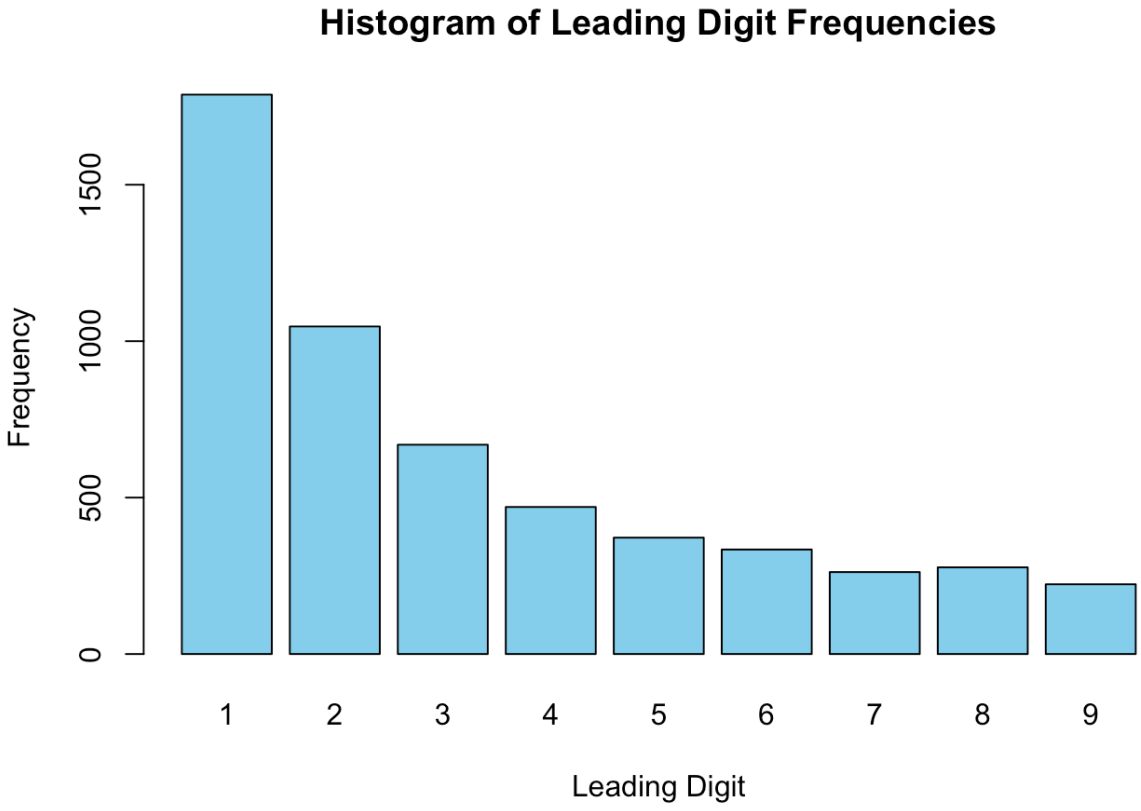


Total Revenues

```
Benford1906_Revenues_total<-Benford1906$Revenues_Total
leading_digits_1906_Revenues_total<-as.numeric(substr(as.character(Benford1906_Revenues_total),1,1))
chisq.benftest(Benford1906_Revenues_total[Benford1906_Revenues_total>0])
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Revenues_total[Benford1906_Revenues_total > 0]
## chisq = 50.729, p-value = 2.959e-08
```

```
digit_frequencies_revenues_total <- table(factor(leading_digits_1906_Revenues_total, levels = 1:9))
barplot(
  digit_frequencies_revenues_total,
  main = "Histogram of Leading Digit Frequencies",
  xlab = "Leading Digit",
  ylab = "Frequency",
  col = "skyblue",
  names.arg = 1:9  # Explicitly set the labels for 1 to 9
)
```



Data in South Australia

```
Benford1906_SA <- subset(Benford1906, State == "SA")
Benford1906_Revenues_total_SA <- Benford1906_SA$Revenues_Total
leading_digits_1906_Revenues_total_SA <- as.numeric(
  substr(as.character(Benford1906_Revenues_total_SA[Benford1906_Revenues_total_SA > 0]), 1, 1)
)
library(BenfordTests)
chisq_result_Revenues_total_SA <- chisq.benftest(Benford1906_Revenues_total_SA[Benford1906_Revenues_to
tal_SA > 0])
chisq_result_Revenues_total_SA
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Revenues_total_SA[Benford1906_Revenues_total_SA >     0]
## chisq = 19.413, p-value = 0.0128
```

=> the p-value is large

```
Benford1906_Posted_letters_SA <- Benford1906_SA$Posted_Letters
leading_digits_1906_Posted_letters_SA <- as.numeric(
  substr(as.character(Benford1906_Posted_letters_SA[Benford1906_Posted_letters_SA > 0]), 1, 1)
)
library(BenfordTests)
chisq_result_Posted_letters_SA <- chisq.benftest(Benford1906_Posted_letters_SA[Benford1906_Posted_lett
ers_SA > 0])
chisq_result_Posted_letters_SA
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Posted_letters_SA[Benford1906_Posted_letters_SA >     0]
## chisq = 8.4356, p-value = 0.3921
```

NSW

```
Benford1906_NSW <- subset(Benford1906, State == "NSW")
Benford1906_Posted_letters_NSW <- Benford1906_NSW$Posted_Letters
leading_digits_1906_Posted_letters_NSW <- as.numeric(
  substr(as.character(Benford1906_Posted_letters_NSW[Benford1906_Posted_letters_NSW > 0]), 1, 1)
)
library(BenfordTests)
chisq_result_Posted_letters_NSW <- chisq.benftest(Benford1906_Posted_letters_NSW[Benford1906_Posted_le
tters_SA > 0])
chisq_result_Posted_letters_SA
```

```
##
##  Chi-Square Test for Benford Distribution
##
## data:  Benford1906_Posted_letters_SA[Benford1906_Posted_letters_SA >     0]
## chisq = 8.4356, p-value = 0.3921
```

Extract Population Data and Compute Leading Digits

```
# Remove non-numeric values from Population
Benford1906_Population <- Benford1906$Population
Benford1913_Population <- Benford1913$Population

# Get leading digits
leading_digits_1906_Population <- as.numeric(substr(as.character(Benford1906_Population), 1, 1))
leading_digits_1913_Population <- as.numeric(substr(as.character(Benford1913_Population), 1, 1))

# Calculate the frequency of each leading digit (1 to 9)
digit_frequencies_1906 <- table(factor(leading_digits_1906_Population, levels = 1:9))
digit_frequencies_1913 <- table(factor(leading_digits_1913_Population, levels = 1:9))
```

Benford’s Expected Distribution

```
benford_expected <- data.frame(
  LeadingDigit = 1:9,
  Percent = c(30.1, 17.6, 12.5, 9.7, 7.9, 6.7, 5.8, 5.1, 4.6)
)
```

Combined Subplots: Histogram & Bar-Line Chart

```
library(ggplot2)
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
```

```
library(gridExtra)
```

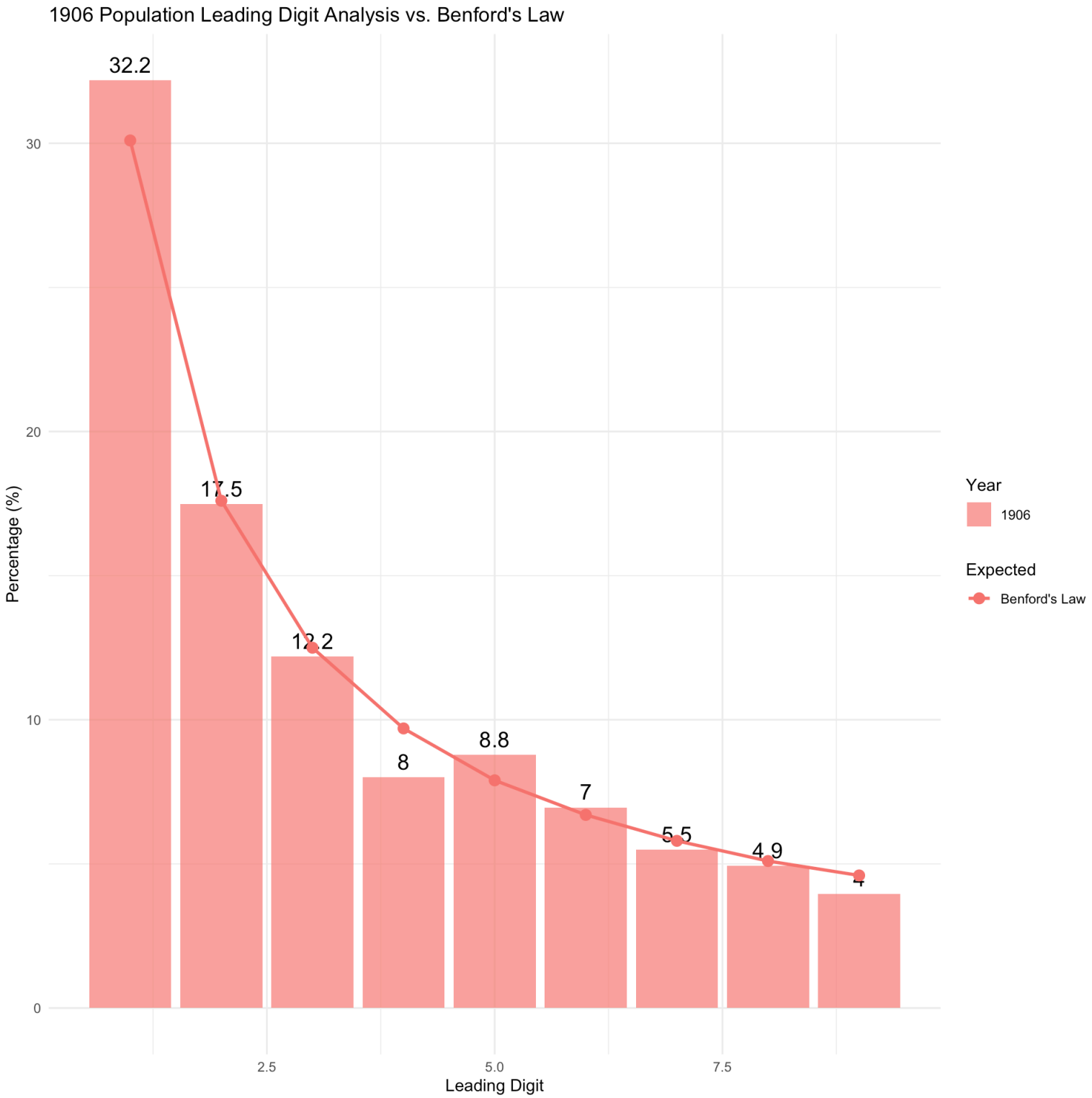
```
##
## Attaching package: 'gridExtra'
```

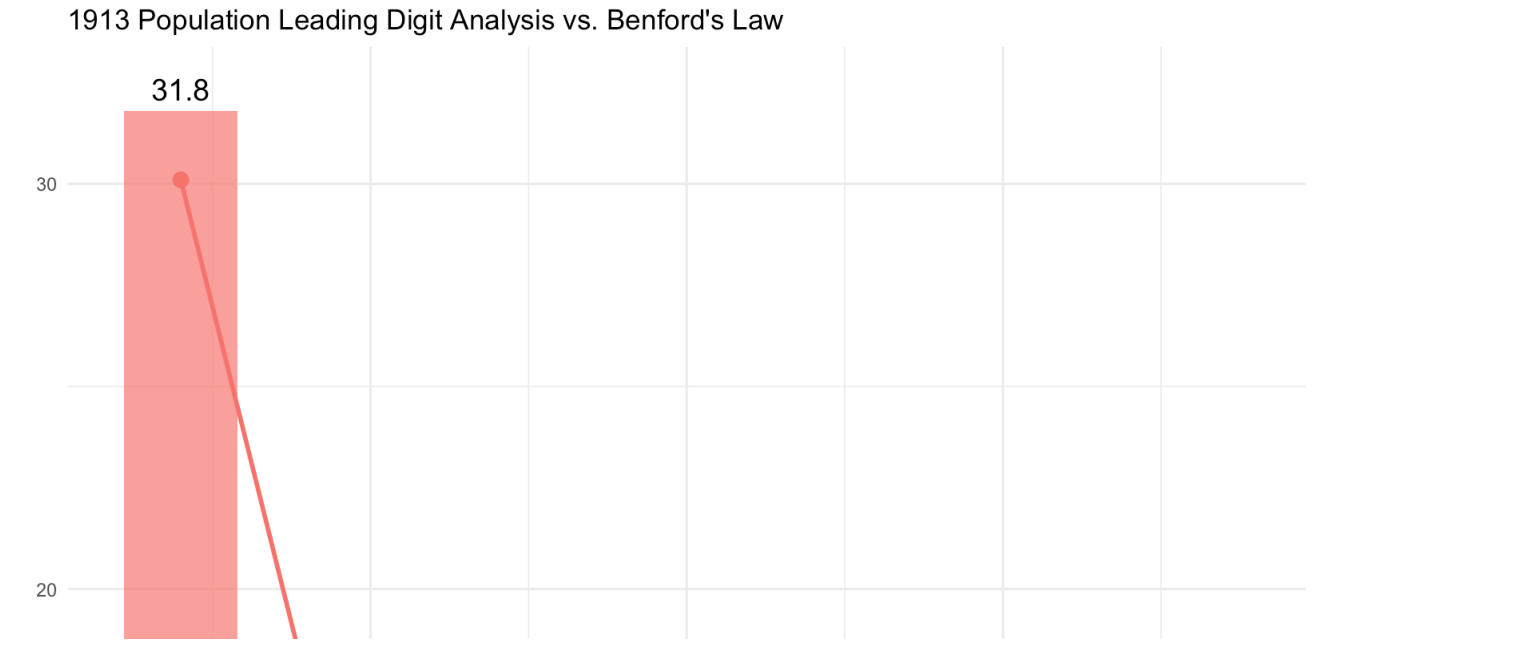
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

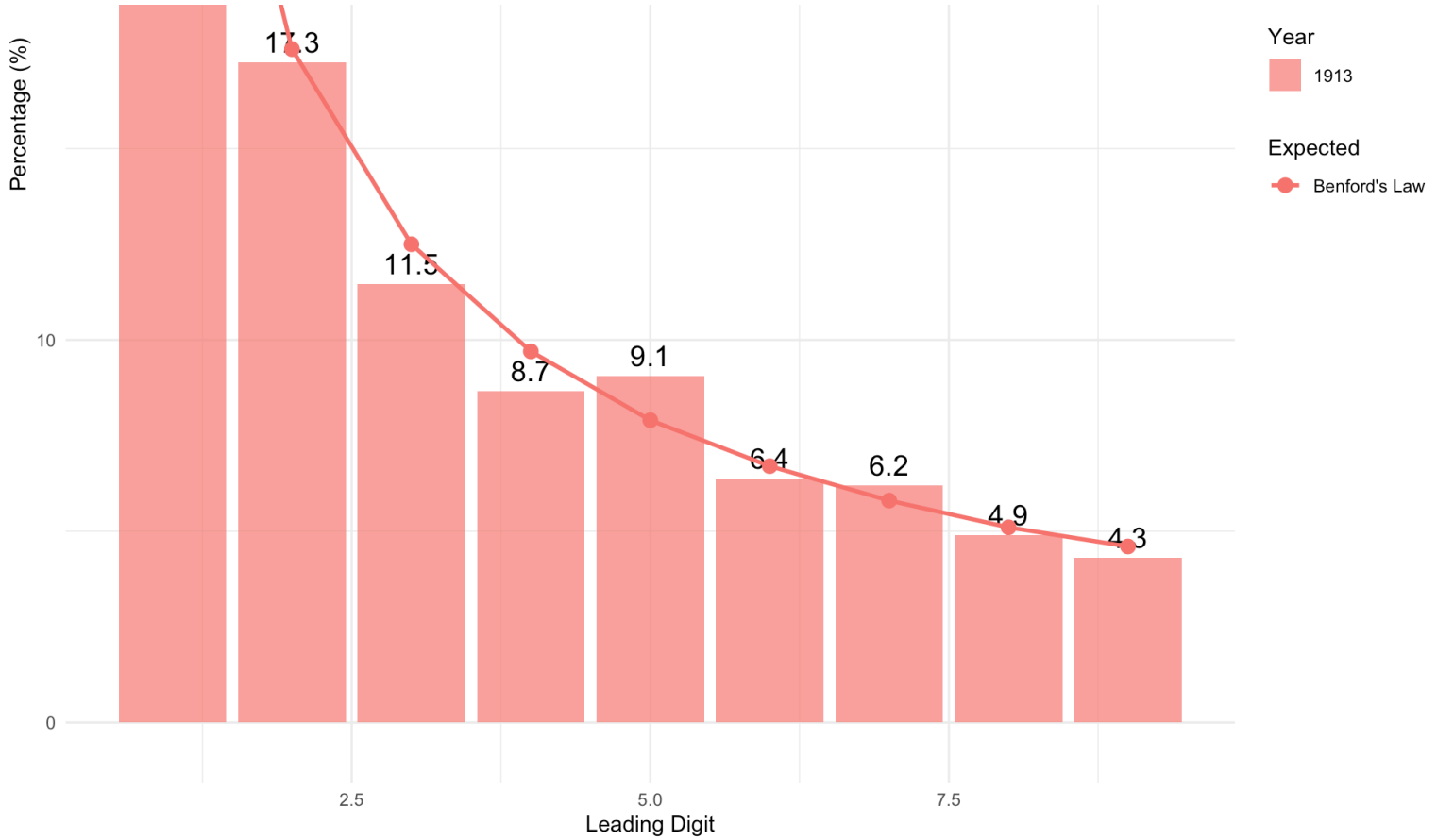
```
# Convert frequency to percentage
digit_frequencies_1906_df <- data.frame(
  LeadingDigit = 1:9,
  Percent = as.numeric(digit_frequencies_1906) / sum(digit_frequencies_1906) * 100
)
digit_frequencies_1913_df <- data.frame(
  LeadingDigit = 1:9,
  Percent = as.numeric(digit_frequencies_1913) / sum(digit_frequencies_1913) * 100
)

# Histogram Plots with Value Labels
hist_1906 <- ggplot(digit_frequencies_1906_df, aes(x = LeadingDigit, y = Percent)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = round(Percent, 1)), vjust = -0.5, size = 5) +
  labs(title = "Histogram (1906 Population)", x = "Leading Digit", y = "Frequency (%)") +
  theme_minimal()

hist_1913 <- ggplot(digit_frequencies_1913_df, aes(x = LeadingDigit, y = Percent)) +
  geom_bar(stat = "identity", fill = "lightcoral") +
  geom_text(aes(label = round(Percent, 1)), vjust = -0.5, size = 5) +
  labs(title = "Histogram (1913 Population)", x = "Leading Digit", y = "Frequency (%)") +
  theme_minimal()

# Bar-Line Chart for 1906
bar_line_chart_1906 <- ggplot() +
  geom_bar(data = digit_frequencies_1906_df, aes(x = LeadingDigit, y = Percent, fill = "1906"), stat =
"identity", alpha = 0.7) +
  geom_text(data = digit_frequencies_1906_df, aes(x = LeadingDigit, y = Percent, label = round(Percen
t, 1)), vjust = -0.5, size = 5) +
  geom_line(data = benford_expected, aes(x = LeadingDigit, y = Percent, color = "Benford's Law"), size
= 1) +
  geom_point(data = benford_expected, aes(x = LeadingDigit, y = Percent, color = "Benford's Law"), siz
e = 3) +
  labs(title = "1906 Population Leading Digit Analysis vs. Benford's Law",
       x = "Leading Digit",
       y = "Percentage (%)",
       fill = "Year",
       color = "Expected") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

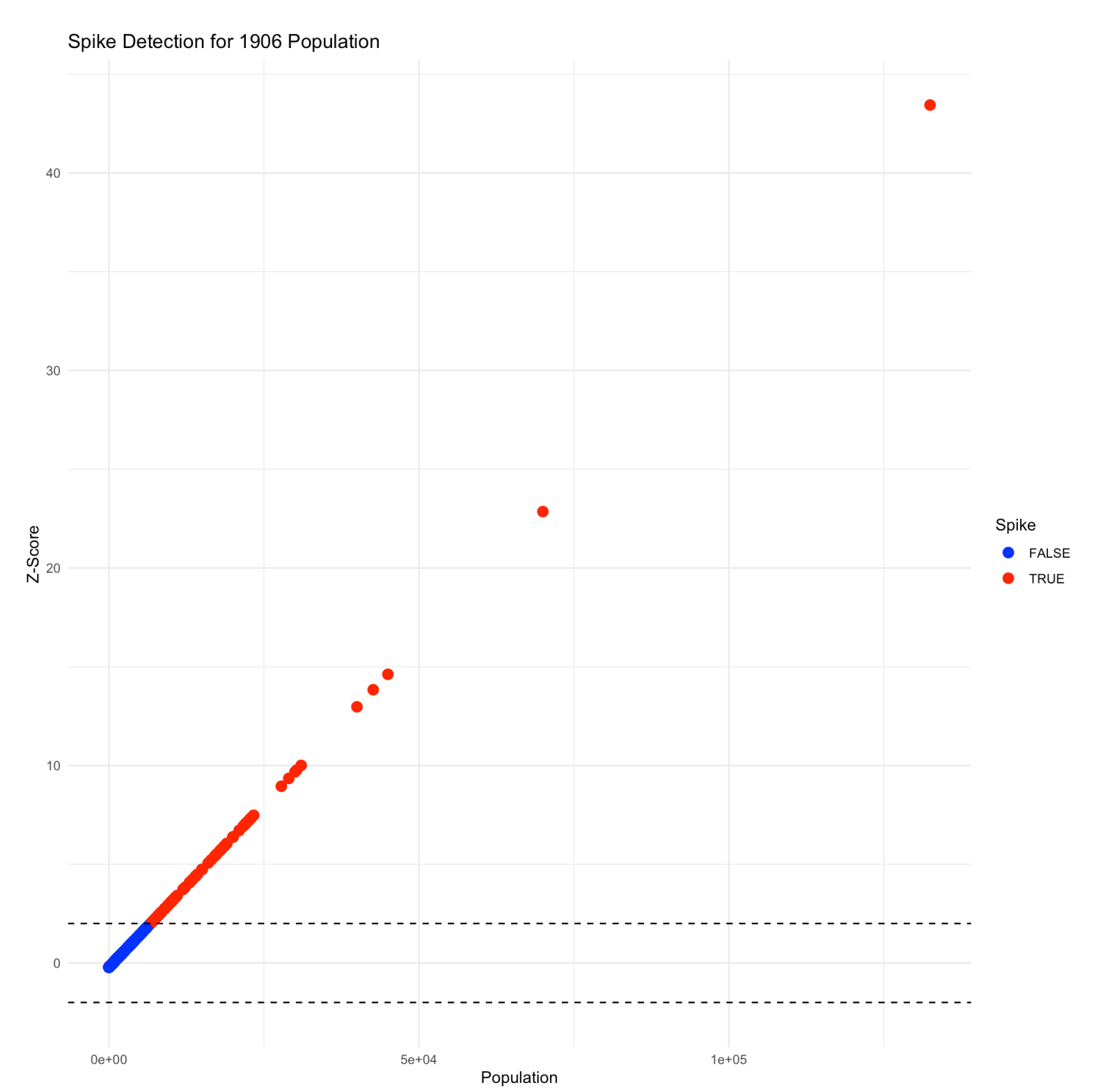
```
# Bar-Line Chart for 1913
bar_line_chart_1913 <- ggplot() +
  geom_bar(data = digit_frequencies_1913_df, aes(x = LeadingDigit, y = Percent, fill = "1913"), stat =
"identity", alpha = 0.7) +
  geom_text(data = digit_frequencies_1913_df, aes(x = LeadingDigit, y = Percent, label = round(Percen
t, 1)), vjust = -0.5, size = 5) +
  geom_line(data = benford_expected, aes(x = LeadingDigit, y = Percent, color = "Benford's Law"), size
= 1) +
  geom_point(data = benford_expected, aes(x = LeadingDigit, y = Percent, color = "Benford's Law"), siz
e = 3) +
  labs(title = "1913 Population Leading Digit Analysis vs. Benford's Law",
       x = "Leading Digit",
       y = "Percentage (%)",
       fill = "Year",
       color = "Expected") +
  theme_minimal()


# Detect Spikes (values significantly deviating from Benford's Law)
spike_threshold <- 5  # Define a threshold for significant deviation
spikes_1906 <- digit_frequencies_1906_df %>% filter(abs(Percent - benford_expected$Percent) > spike_th
reshold)
spikes_1913 <- digit_frequencies_1913_df %>% filter(abs(Percent - benford_expected$Percent) > spike_th
reshold)

# Highlight Spikes in Bar-Line Charts
bar_line_chart_1906 <- bar_line_chart_1906 +
  geom_point(data = spikes_1906, aes(x = LeadingDigit, y = Percent), color = "red", size = 4, shape =
8)

bar_line_chart_1913 <- bar_line_chart_1913 +
  geom_point(data = spikes_1913, aes(x = LeadingDigit, y = Percent), color = "red", size = 4, shape =
8)

# Arrange plots in a 2x2 layout
grid.arrange(bar_line_chart_1906, bar_line_chart_1913, ncol = 1, nrow = 2)
```

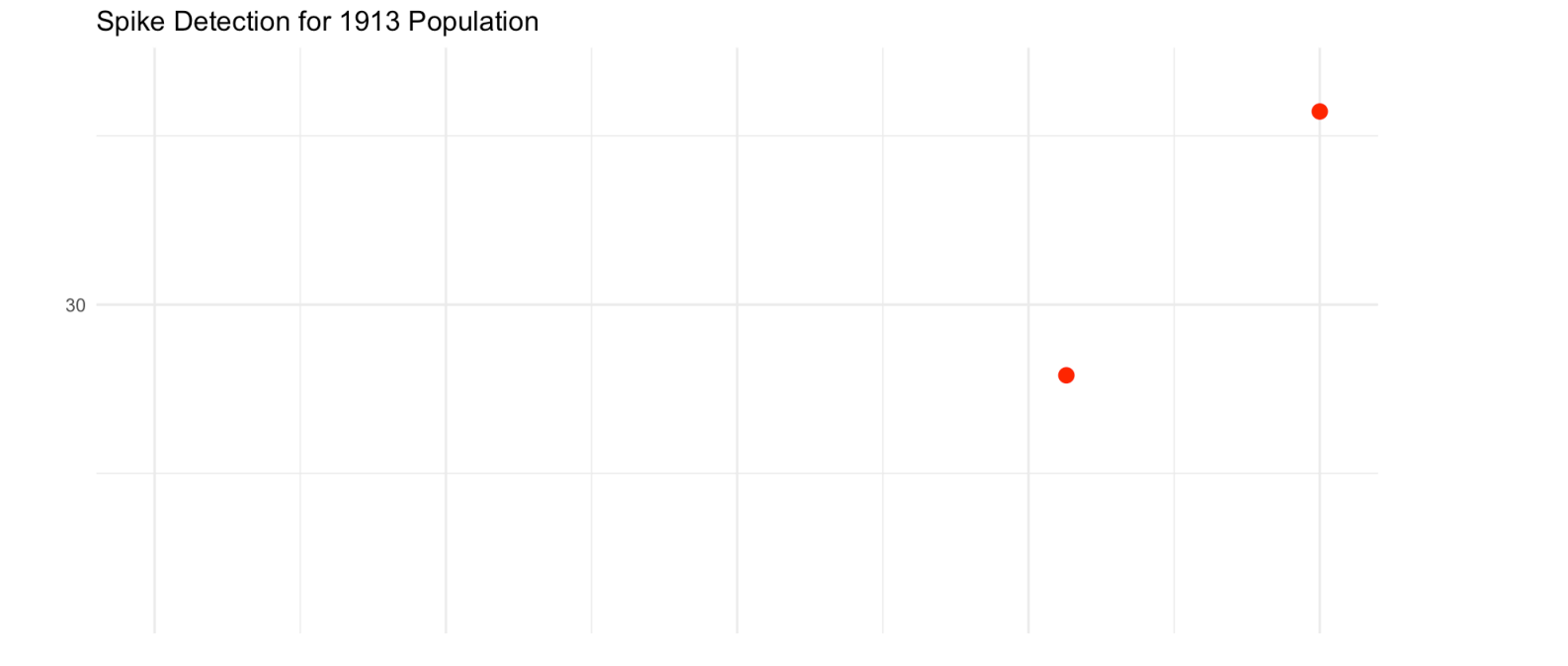

1906 Population Leading Digit Analysis vs. Benford's Law
32.2
17.5
12.2
8
8.8
7
5.5
4.9
4
30
20
10
0
Percentage (%)
2.5
5.0
7.5
Leading Digit
Year
1906
Expected
Benford's Law

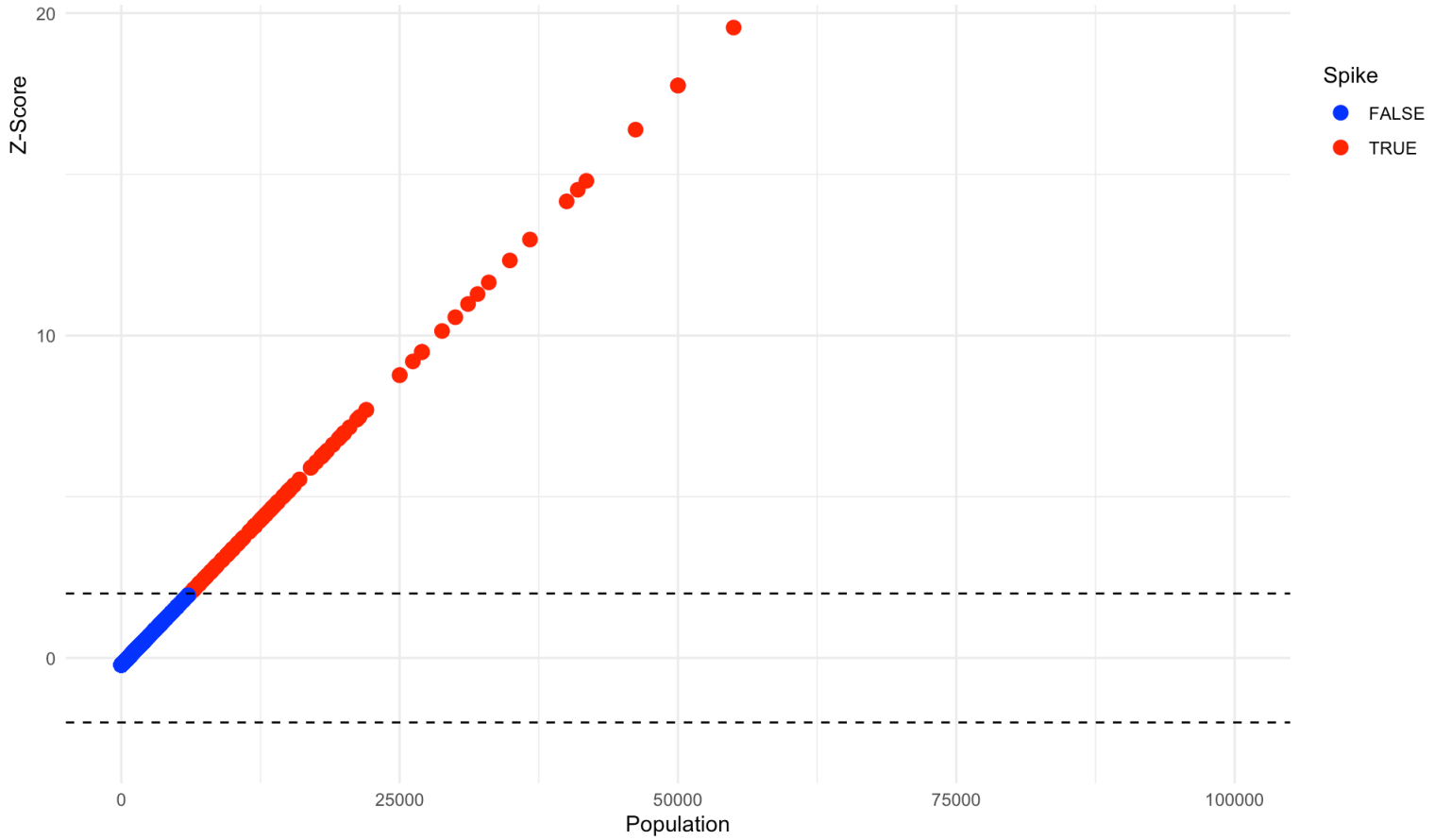

1913 Population Leading Digit Analysis vs. Benford's Law
31.8
30
20




```r
library(ggplot2)
library(dplyr)
library(gridExtra)

#  z-score function to detect spike
detect_spikes <- function(population_data, year) {
  population_data <- na.omit(population_data)
  mean_pop <- mean(population_data)
  sd_pop <- sd(population_data)

  #  z-score
  z_scores <- (population_data - mean_pop) / sd_pop

  # spike: |z-score| > 2
  spike_data <- data.frame(Population = population_data, Z_Score = z_scores,
                           Spike = abs(z_scores) > 2)

  # plot
  spike_plot <- ggplot(spike_data, aes(x = Population, y = Z_Score, color = Spike)) +
    geom_point(size = 3) +
    scale_color_manual(values = c("FALSE" = "blue", "TRUE" = "red")) +
    geom_hline(yintercept = c(-2, 2), linetype = "dashed", color = "black") +
    labs(title = paste("Spike Detection for", year, "Population"),
         x = "Population", y = "Z-Score") +
    theme_minimal()

  return(spike_plot)
}

# 1906 and 1913 dataset
spike_plot_1906 <- detect_spikes(Benford1906$Population, "1906")
spike_plot_1913 <- detect_spikes(Benford1913$Population, "1913")

# show
grid.arrange(spike_plot_1906, spike_plot_1913, ncol = 1, nrow = 2, heights = c(1,1))
```


Spike Detection for 1906 Population
Z-Score
40
30
20
10
0
0e+00
5e+04
1e+05
Population
Spike
FALSE
TRUE


Spike Detection for 1913 Population
30


Z-Score
20
10
0
0
25000
50000
75000
100000
Population
Spike
FALSE
TRUE