BIN381_Project_Milestone

Group F

2024-10-09

R Markdown

```
#Packages to Install
#install.packages("e1071")
#install.packages("lubridate")
#install.packages("ggplot2")
#install.packages("reshape2")
```

##DATA SELECTION

```
# Read 'CustData2.csv' file into data frame 'customers'
customers <- read.csv("CustData2.csv")

# Display structure of the data frame
str(customers)
```

```
## 'data.frame':    191323 obs. of  24 variables:
##  $ Column1                           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Last.Name                         : chr  "ALBERT" "ARGUELLO"
"TUCKER" "DELL" ...
##  $ First.Name                        : chr  "JESSICA" "ADRIAN" "KEVIN"
"JAMES" ...
##  $ Middle.Initial                    : chr  "M" "A" "K" "A" ...
##  $ Title                             : chr  "CORRECTIONAL OFFICER"
"POLICE OFFICER" "CORRECTIONAL OFFICER" "WASTE SCALE OPERATOR" ...
##  $ Department.Name                   : chr  "CORRECTIONS &
REHABILITATION" "POLICE" "CORRECTIONS & REHABILITATION" "SOLID WASTE
MANAGEMENT" ...
##  $ Annual.Salary                     : num  54620 65250 62394 37735
64386 ...
##  $ Gross.Pay.Last.Paycheck           : num  2502 3468 4514 1562 6666
...
##  $ Gross.Year.To.Date                : num  48025 57932 49968 35470
132851 ...
##  $ Gross.Year.To.Date...FRS.Contribution: num  46617 56223 48501 34433
128949 ...
##  $ year_of_birth                     : int  1976 1964 1942 1977 1949
1950 1946 1978 1949 1951 ...
##  $ marital_status                    : chr  "married" "" "single"
"married" ...
##  $ street_address                    : chr  "27 North Sagadahoc
Boulevard" "37 West Geneva Street" "47 Toa Alta Road" "47 South Kanabec Road"
...
##  $ postal_code                       : int  60332 55406 34077 72996
```

```
67644 83786 52773 37400 71349 55056 ...
##  $ city                                : chr  "Ede" "Hoofddorp"
"Schimmert" "Scheveningen" ...
##  $ State                               : chr  "Gelderland" "Noord"
"Limburg" "Zuid" ...
##  $ Province                            : chr  "" "Holland" "" "Holland"
...
##  $ Country_id                          : int  52770 52770 52770 52770
52775 52782 52775 52782 52770 52789 ...
##  $ phone_number                        : chr  "519-236-6123" "327-194-
5008" "288-613-9676" "222-269-1259" ...
##  $ email                               : chr  "Ruddy@company.com"
"Ruddy@company.com" "Ruddy@company.com" "Ruddy@company.com" ...
##  $ Education                           : chr  "Masters" "Masters"
"Masters" "Masters" ...
##  $ Occupation                          : chr  "Prof." "Prof." "Prof."
"Prof." ...
##  $ household_size                      : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ yrs_residence                       : int  4 4 4 4 4 4 4 4 4 4 ...
```

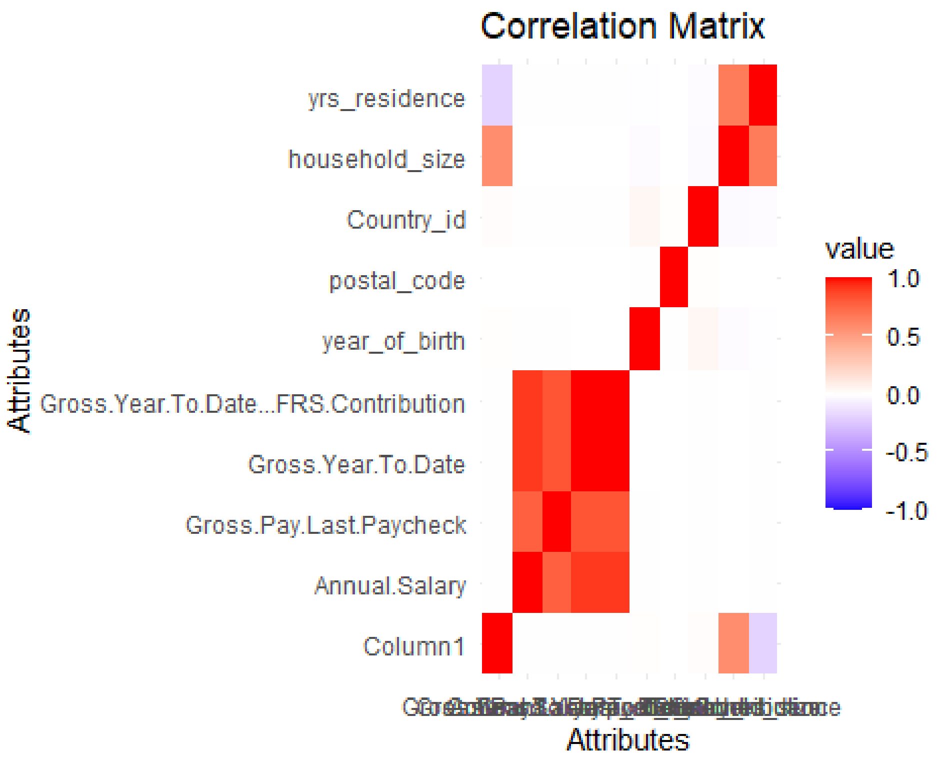
```
#Import libraries for plotting
library(ggplot2)
library(reshape2)

# Select numerical attributes
numeric_data <- customers[sapply(customers, is.numeric)]

# Calculate correlation matrix
correlation_matrix <- cor(numeric_data, use = "complete.obs")
print(correlation_matrix)
```

```
##                                             Column1 Annual.Salary
## Column1                                1.0000000000 -0.0036675519
## Annual.Salary                         -0.0036675519  1.0000000000
## Gross.Pay.Last.Paycheck               -0.0047217061  0.7772558821
## Gross.Year.To.Date                    -0.0049238819  0.9122270032
## Gross.Year.To.Date...FRS.Contribution -0.0048931111  0.9122753526
## year_of_birth                          0.0071862933 -0.0026621848
## postal_code                           -0.0005331626  0.0005061666
## Country_id                             0.0138730870  0.0054505876
## household_size                         0.5820135284 -0.0007670503
## yrs_residence                         -0.1888747148  0.0043115974
##                                       Gross.Pay.Last.Paycheck
## Column1                                         -0.0047217061
## Annual.Salary                                    0.7772558821
## Gross.Pay.Last.Paycheck                          1.0000000000
## Gross.Year.To.Date                               0.8224769696
## Gross.Year.To.Date...FRS.Contribution            0.8217490345
## year_of_birth                                   -0.0026137912
## postal_code                                     -0.0009590673
## Country_id                                       0.0039965284
## household_size                                  -0.0013831223
## yrs_residence                                    0.0046397673
##                                       Gross.Year.To.Date
## Column1                                     -0.004923882
## Annual.Salary                                0.912227003
## Gross.Pay.Last.Paycheck                      0.822476970
## Gross.Year.To.Date                           1.000000000
## Gross.Year.To.Date...FRS.Contribution        0.999835351
## year_of_birth                               -0.001644027
## postal_code                                  0.001628696
## Country_id                                   0.005658527
## household_size                              -0.001136563
## yrs_residence                                0.005453532
## 
Gross.Year.To.Date...FRS.Contribution
## Column1                                                        -
0.004893111
## Annual.Salary
0.912275353
## Gross.Pay.Last.Paycheck
0.821749035
## Gross.Year.To.Date
0.999835351
## Gross.Year.To.Date...FRS.Contribution
1.000000000
## year_of_birth                                                  -
0.001699777
## postal_code
0.001618253
## Country_id
0.005630730
## household_size                                                 -
0.001086514
## yrs_residence
0.005489229
##                                       year_of_birth   postal_code
Country_id
## Column1                                 0.007186293 -0.0005331626
0.013873087
## Annual.Salary                          -0.002662185  0.0005061666
0.005450588
## Gross.Pay.Last.Paycheck                -0.002613791 -0.0009590673
0.003996528
## Gross.Year.To.Date                     -0.001644027  0.0016286961
0.005658527
## Gross.Year.To.Date...FRS.Contribution  -0.001699777  0.0016182533
0.005630730
## year_of_birth                           1.000000000 -0.0044900811
0.042904593
```

```
## postal_code                             -0.004490081  1.0000000000
0.005828755
## Country_id                               0.042904593  0.0058287550
1.000000000
## household_size                          -0.015288080  0.0017671756 -
0.023520125
## yrs_residence                           -0.010114024  0.0011539062 -
0.015541244
##                                        household_size yrs_residence
## Column1                                  0.5820135284  -0.188874715
## Annual.Salary                           -0.0007670503   0.004311597
## Gross.Pay.Last.Paycheck                 -0.0013831223   0.004639767
## Gross.Year.To.Date                      -0.0011365634   0.005453532
## Gross.Year.To.Date...FRS.Contribution  -0.0010865140   0.005489229
## year_of_birth                           -0.0152880799  -0.010114024
## postal_code                              0.0017671756   0.001153906
## Country_id                              -0.0235201249  -0.015541244
## household_size                           1.0000000000   0.661607624
## yrs_residence                            0.6616076237   1.000000000
```

```
melted_corr_matrix <- melt(correlation_matrix)
```

```
ggplot(data = melted_corr_matrix, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint =
0, limit = c(-1, 1)) +
  theme_minimal() +
  labs(title = "Correlation Matrix", x = "Attributes", y = "Attributes")
```



```
# ** Cardinality **
# Create a function to calculate the cardinality (number of unique values)
calculate_cardinality <- function(df) {
  cardinalities <- sapply(df, function(x) length(unique(x)))
  return(cardinalities)
}

# Calculate the cardinality for each attribute in the dataset
cardinality <- calculate_cardinality(customers)

# Display the cardinality of each attribute
print("Cardinality (number of unique values) for each attribute:")

## [1] "Cardinality (number of unique values) for each attribute:"

print(cardinality)

##                                Column1
Last.Name
##                                 191323
10917
##                             First.Name
Middle.Initial
##                                   7235
27
##                                  Title
Department.Name
```

```
##                                    2291
43
##                           Annual.Salary
Gross.Pay.Last.Paycheck
##                                    3996
16180
##                      Gross.Year.To.Date
Gross.Year.To.Date...FRS.Contribution
##                                   27096
27321
##                           year_of_birth
marital_status
##                                      75
12
##                          street_address
postal_code
##                                   50945
623
##                                    city
State
##                                     614
142
##                                Province
Country_id
##                                      31
19
##                            phone_number
email
##                                   51000
1699
##                               Education
Occupation
##                                       3
4
##                          household_size
yrs_residence
##                                       2
4

# Create a table or dataframe for better visualization
cardinality_df <- data.frame(Attribute = names(cardinality), Cardinality =
cardinality)

#Sort the results by cardinality to easily identify attributes with high or
low cardinality
cardinality_df <- cardinality_df[order(-cardinality_df$Cardinality),]

# Print the sorted cardinality dataframe
print(cardinality_df)
```

```
##
Attribute
## Column1
Column1
## phone_number
phone_number
## street_address
street_address
## Gross.Year.To.Date...FRS.Contribution
Gross.Year.To.Date...FRS.Contribution
## Gross.Year.To.Date
Gross.Year.To.Date
## Gross.Pay.Last.Paycheck
Gross.Pay.Last.Paycheck
## Last.Name
Last.Name
## First.Name
First.Name
## Annual.Salary
Annual.Salary
## Title
Title
## email
email
## postal_code
postal_code
## city
city
## State
State
## year_of_birth
year_of_birth
## Department.Name
Department.Name
## Province
Province
## Middle.Initial
Middle.Initial
## Country_id
Country_id
## marital_status
marital_status
## Occupation
Occupation
## yrs_residence
yrs_residence
## Education
Education
## household_size
household_size
```

```
##                                       Cardinality
## Column1                                    191323
## phone_number                                51000
## street_address                              50945
## Gross.Year.To.Date...FRS.Contribution       27321
## Gross.Year.To.Date                          27096
## Gross.Pay.Last.Paycheck                     16180
## Last.Name                                   10917
## First.Name                                   7235
## Annual.Salary                                3996
## Title                                        2291
## email                                        1699
## postal_code                                   623
## city                                          614
## State                                         142
## year_of_birth                                  75
## Department.Name                                43
## Province                                       31
## Middle.Initial                                 27
## Country_id                                     19
## marital_status                                 12
## Occupation                                      4
## yrs_residence                                   4
## Education                                       3
## household_size                                  2
```

DATA CLEANING

```
# Display structure of the data frame
str(customers)
```

```
## 'data.frame':    191323 obs. of  24 variables:
##  $ Column1                              : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Last.Name                            : chr  "ALBERT" "ARGUELLO"
"TUCKER" "DELL" ...
##  $ First.Name                           : chr  "JESSICA" "ADRIAN" "KEVIN"
"JAMES" ...
##  $ Middle.Initial                       : chr  "M" "A" "K" "A" ...
##  $ Title                                : chr  "CORRECTIONAL OFFICER"
"POLICE OFFICER" "CORRECTIONAL OFFICER" "WASTE SCALE OPERATOR" ...
##  $ Department.Name                      : chr  "CORRECTIONS &
REHABILITATION" "POLICE" "CORRECTIONS & REHABILITATION" "SOLID WASTE
MANAGEMENT" ...
##  $ Annual.Salary                        : num  54620 65250 62394 37735
64386 ...
##  $ Gross.Pay.Last.Paycheck              : num  2502 3468 4514 1562 6666
...
##  $ Gross.Year.To.Date                   : num  48025 57932 49968 35470
132851 ...
##  $ Gross.Year.To.Date...FRS.Contribution: num  46617 56223 48501 34433
```

```
128949 ...
##  $ year_of_birth                        : int  1976 1964 1942 1977 1949
1950 1946 1978 1949 1951 ...
##  $ marital_status                       : chr  "married" "" "single"
"married" ...
##  $ street_address                       : chr  "27 North Sagadahoc
Boulevard" "37 West Geneva Street" "47 Toa Alta Road" "47 South Kanabec Road"
...
##  $ postal_code                          : int  60332 55406 34077 72996
67644 83786 52773 37400 71349 55056 ...
##  $ city                                 : chr  "Ede" "Hoofddorp"
"Schimmert" "Scheveningen" ...
##  $ State                                : chr  "Gelderland" "Noord"
"Limburg" "Zuid" ...
##  $ Province                             : chr  "" "Holland" "" "Holland"
...
##  $ Country_id                           : int  52770 52770 52770 52770
52775 52782 52775 52782 52770 52789 ...
##  $ phone_number                         : chr  "519-236-6123" "327-194-
5008" "288-613-9676" "222-269-1259" ...
##  $ email                                : chr  "Ruddy@company.com"
"Ruddy@company.com" "Ruddy@company.com" "Ruddy@company.com" ...
##  $ Education                            : chr  "Masters" "Masters"
"Masters" "Masters" ...
##  $ Occupation                           : chr  "Prof." "Prof." "Prof."
"Prof." ...
##  $ household_size                       : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ yrs_residence                        : int  4 4 4 4 4 4 4 4 4 4 ...
```

```
# Import 'lubridate' package to work with Date types
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
# Create a new column/attribute that calculates the customers age based on
'year of birth'
customers$Age <- as.integer(year(today()) - customers$year_of_birth)

# Display structure of the data frame
str(customers)
```

```
## 'data.frame':    191323 obs. of  25 variables:
##  $ Column1                              : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Last.Name                            : chr  "ALBERT" "ARGUELLO"
"TUCKER" "DELL" ...
##  $ First.Name                           : chr  "JESSICA" "ADRIAN" "KEVIN"
```

```
"JAMES" ...
##  $ Middle.Initial                       : chr  "M" "A" "K" "A" ...
##  $ Title                                : chr  "CORRECTIONAL OFFICER"
"POLICE OFFICER" "CORRECTIONAL OFFICER" "WASTE SCALE OPERATOR" ...
##  $ Department.Name                      : chr  "CORRECTIONS &
REHABILITATION" "POLICE" "CORRECTIONS & REHABILITATION" "SOLID WASTE
MANAGEMENT" ...
##  $ Annual.Salary                        : num  54620 65250 62394 37735
64386 ...
##  $ Gross.Pay.Last.Paycheck              : num  2502 3468 4514 1562 6666
...
##  $ Gross.Year.To.Date                   : num  48025 57932 49968 35470
132851 ...
##  $ Gross.Year.To.Date...FRS.Contribution: num  46617 56223 48501 34433
128949 ...
##  $ year_of_birth                        : int  1976 1964 1942 1977 1949
1950 1946 1978 1949 1951 ...
##  $ marital_status                       : chr  "married" "" "single"
"married" ...
##  $ street_address                       : chr  "27 North Sagadahoc
Boulevard" "37 West Geneva Street" "47 Toa Alta Road" "47 South Kanabec Road"
...
##  $ postal_code                          : int  60332 55406 34077 72996
67644 83786 52773 37400 71349 55056 ...
##  $ city                                 : chr  "Ede" "Hoofddorp"
"Schimmert" "Scheveningen" ...
##  $ State                                : chr  "Gelderland" "Noord"
"Limburg" "Zuid" ...
##  $ Province                             : chr  "" "Holland" "" "Holland"
...
##  $ Country_id                           : int  52770 52770 52770 52770
52775 52782 52775 52782 52770 52789 ...
##  $ phone_number                         : chr  "519-236-6123" "327-194-
5008" "288-613-9676" "222-269-1259" ...
##  $ email                                : chr  "Ruddy@company.com"
"Ruddy@company.com" "Ruddy@company.com" "Ruddy@company.com" ...
##  $ Education                            : chr  "Masters" "Masters"
"Masters" "Masters" ...
##  $ Occupation                           : chr  "Prof." "Prof." "Prof."
"Prof." ...
##  $ household_size                       : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ yrs_residence                        : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Age                                  : int  48 60 82 47 75 74 78 46 75
73 ...

# Create vector with all columns/attributes that need to be kept
keepColumns <- c("Title", "Department.Name", "Annual.Salary",
                 "Gross.Pay.Last.Paycheck", "Gross.Year.To.Date",
                 "Gross.Year.To.Date...FRS.Contribution",
                 "Age", "marital_status", "Country_id", "Education",
                 "Occupation", "household_size", "yrs_residence")

# Remove irrelevant columns/attributes by keeping relevant ones
customers <- customers[keepColumns]

# Display structure of the data frame
str(customers)
```

```
## 'data.frame':    191323 obs. of  13 variables:
##  $ Title                                : chr  "CORRECTIONAL OFFICER"
"POLICE OFFICER" "CORRECTIONAL OFFICER" "WASTE SCALE OPERATOR" ...
##  $ Department.Name                      : chr  "CORRECTIONS &
REHABILITATION" "POLICE" "CORRECTIONS & REHABILITATION" "SOLID WASTE
MANAGEMENT" ...
##  $ Annual.Salary                        : num  54620 65250 62394 37735
64386 ...
##  $ Gross.Pay.Last.Paycheck              : num  2502 3468 4514 1562 6666
...
##  $ Gross.Year.To.Date                   : num  48025 57932 49968 35470
132851 ...
##  $ Gross.Year.To.Date...FRS.Contribution: num  46617 56223 48501 34433
128949 ...
##  $ Age                                  : int  48 60 82 47 75 74 78 46 75
73 ...
##  $ marital_status                       : chr  "married" "" "single"
"married" ...
##  $ Country_id                           : int  52770 52770 52770 52770
52775 52782 52775 52782 52770 52789 ...
##  $ Education                            : chr  "Masters" "Masters"
"Masters" "Masters" ...
##  $ Occupation                           : chr  "Prof." "Prof." "Prof."
"Prof." ...
##  $ household_size                       : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ yrs_residence                        : int  4 4 4 4 4 4 4 4 4 4 ...
```

```
# Cleaning "marital_status"
# Display all of the unique values contained in the 'marital_status'
column/attribute
unique(customers$marital_status)
```

```
##  [1] "married"  ""         "single"   "divorced" "widow"    "Divorc."
##  [7] "NeverM"   "Married"  "Separ."   "Mabsent"  "Widowed"  "Mar-AF"
```

```
# Count the unique values contained in the 'marital_status' column/attribute
length(unique(customers$marital_status))
```

```
## [1] 12
```

```
# Replace incorrect values for "marital_status"
for (i in 1:nrow(customers)) {
  if (customers$marital_status[i] == "Married") {
    customers$marital_status[i] <- "married"
  } else if (customers$marital_status[i] == "Mar-AF") {
    customers$marital_status[i] <- "married"
  } else if (customers$marital_status[i] == "NeverM") {
    customers$marital_status[i] <- "single"
  } else if (customers$marital_status[i] == "Mabsent") {
    customers$marital_status[i] <- "single"
  } else if (customers$marital_status[i] == "Divorc.") {
    customers$marital_status[i] <- "divorced"
  } else if (customers$marital_status[i] == "Separ.") {
    customers$marital_status[i] <- "divorced"
  } else if (customers$marital_status[i] == "widow") {
    customers$marital_status[i] <- "widowed"
  } else if (customers$marital_status[i] == "Widowed") {
    customers$marital_status[i] <- "widowed"
  }
}

# Check to see if "marital_status" was cleaned successfully
unique(customers$marital_status)
```

```
## [1] "married"  ""         "single"   "divorced" "widowed"
```

```
length(unique(customers$marital_status))
```

```
## [1] 5
```

```
# Populating "marital_status"
# Count the number of empty cells
sum(customers$marital_status=="")
```

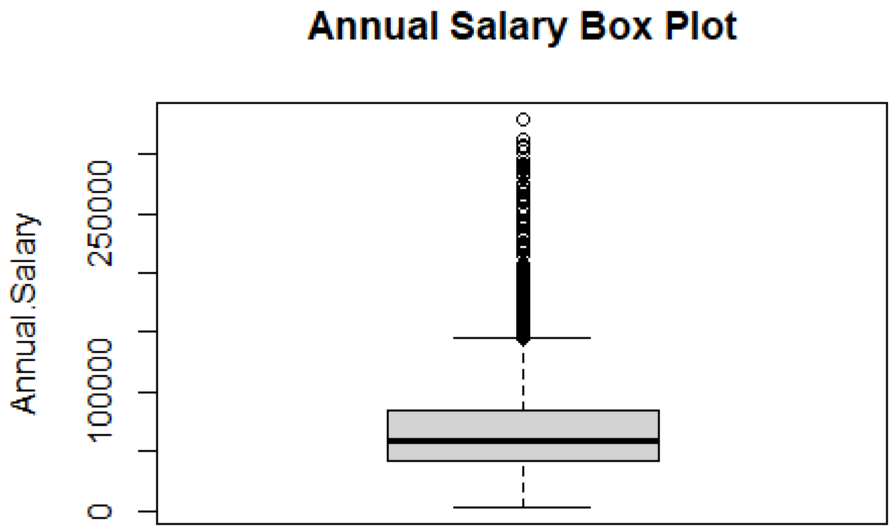
```
## [1] 60795
```

```
# Function to calculate mode
get_mode <- function(v) {
  uniq_vals <- unique(v)
  uniq_vals[which.max(tabulate(match(v, uniq_vals)))]
}

# Get mode value from function
mode_value <-
get_mode(customers$marital_status[!is.na(customers$marital_status) &
                                                  customers$marital_status !=
""])

# Fill missing or empty values in "marital_status" column with mode
customers$marital_status[is.na(customers$marital_status) |
                           customers$marital_status == ""] <- mode_value

# Check if "marital_status" is filled
sum(customers$marital_status=="")
```

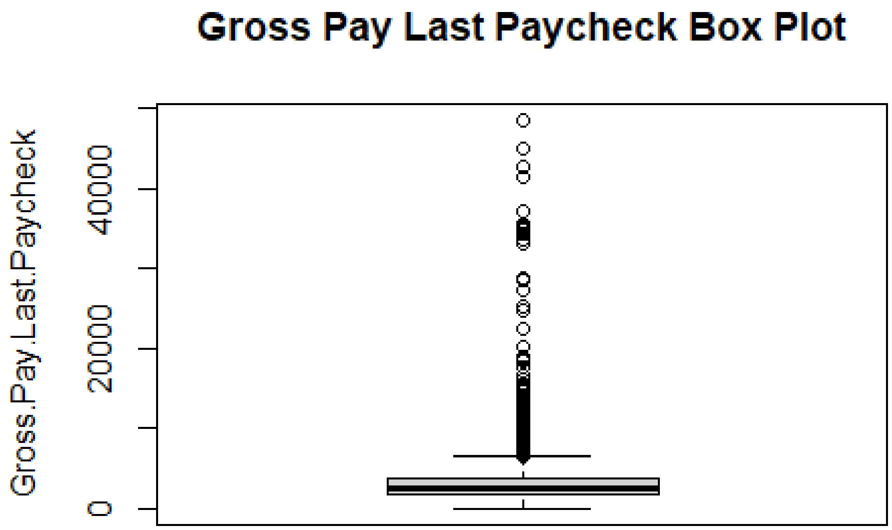
```
## [1] 0

# Missing Values
sum(customers$Title=="")

## [1] 6

sum(customers$Department.Name=="")

## [1] 6

sum(is.na(customers$Annual.Salary))

## [1] 6

sum(is.na(customers$Gross.Pay.Last.Paycheck))

## [1] 6

sum(is.na(customers$Gross.Year.To.Date))

## [1] 6

sum(is.na(customers$Gross.Year.To.Date...FRS.Contribution))

## [1] 6

sum(is.na(customers$Age))

## [1] 0

sum(customers$marital_status=="")

## [1] 0

sum(is.na(customers$Country_id))

## [1] 0

sum(customers$Education=="")

## [1] 0

sum(customers$Occupation=="")

## [1] 0

sum(is.na(customers$household_size))

## [1] 0

sum(is.na(customers$yrs_residence))

## [1] 0
```

```
# Remove empty cells for all columns/attributes
customers <- customers[!(is.na(customers$Title) | customers$Title == "" |
                           is.na(customers$Department.Name)  |
                           customers$Department.Name == ""  |
                           is.na(customers$Annual.Salary)  |
                           customers$Annual.Salary == ""  |
                           is.na(customers$Gross.Pay.Last.Paycheck)  |
                           customers$Gross.Pay.Last.Paycheck == ""  |
                           is.na(customers$Gross.Year.To.Date)  |
                           customers$Gross.Year.To.Date == ""  |

is.na(customers$Gross.Year.To.Date...FRS.Contribution)  |
                           customers$Gross.Year.To.Date...FRS.Contribution ==
""), ]

# Check if there are empty cells left
sum(customers$Title=="")

## [1] 0

sum(customers$Department.Name=="")

## [1] 0

sum(is.na(customers$Annual.Salary))

## [1] 0

sum(is.na(customers$Gross.Pay.Last.Paycheck))

## [1] 0

sum(is.na(customers$Gross.Year.To.Date))

## [1] 0

sum(is.na(customers$Gross.Year.To.Date...FRS.Contribution))

## [1] 0

sum(is.na(customers$Age))

## [1] 0

sum(is.na(customers$Country_id))

## [1] 0

sum(customers$Education=="")

## [1] 0

sum(customers$Occupation=="")
```

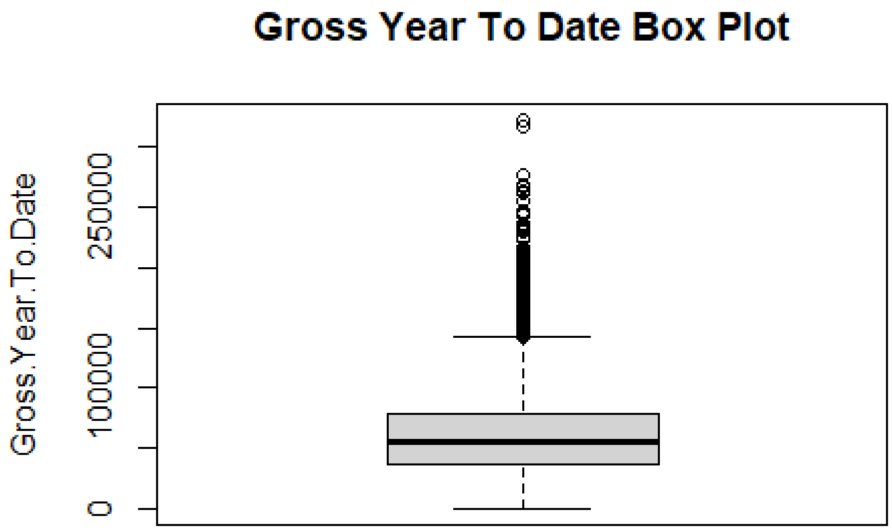
```
## [1] 0

sum(is.na(customers$household_size))

## [1] 0

sum(is.na(customers$yrs_residence))

## [1] 0

# ** Outlier Treatment **
## Display outliers
### Display "Annual.Salary" box plot
boxplot(customers$Annual.Salary,
        main = "Annual Salary Box Plot",
        ylab = "Annual.Salary")
```



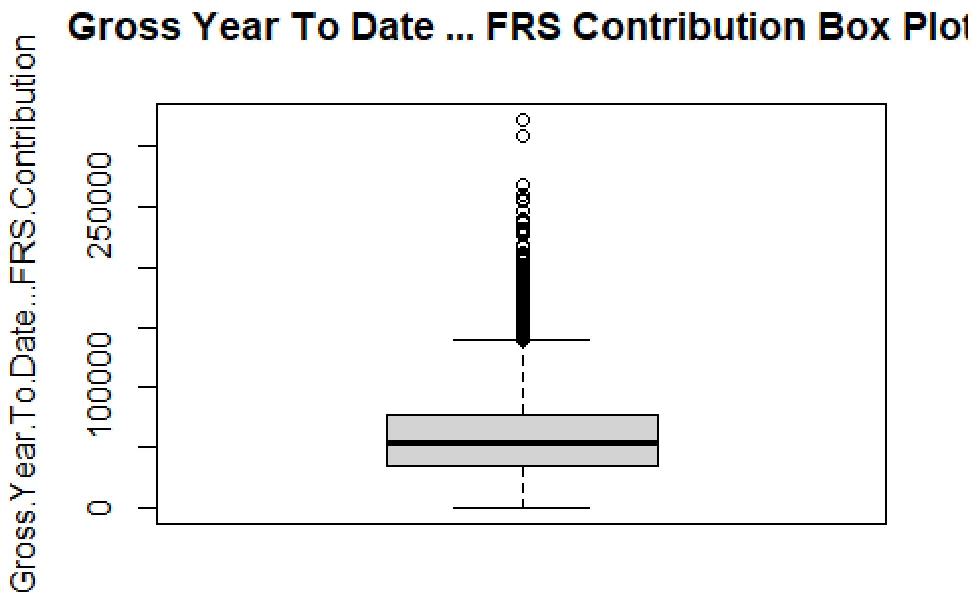
```
### Display "Gross.Pay.Last.Paycheck" box plot
boxplot(customers$Gross.Pay.Last.Paycheck,
        main = "Gross Pay Last Paycheck Box Plot",
        ylab = "Gross.Pay.Last.Paycheck")
```



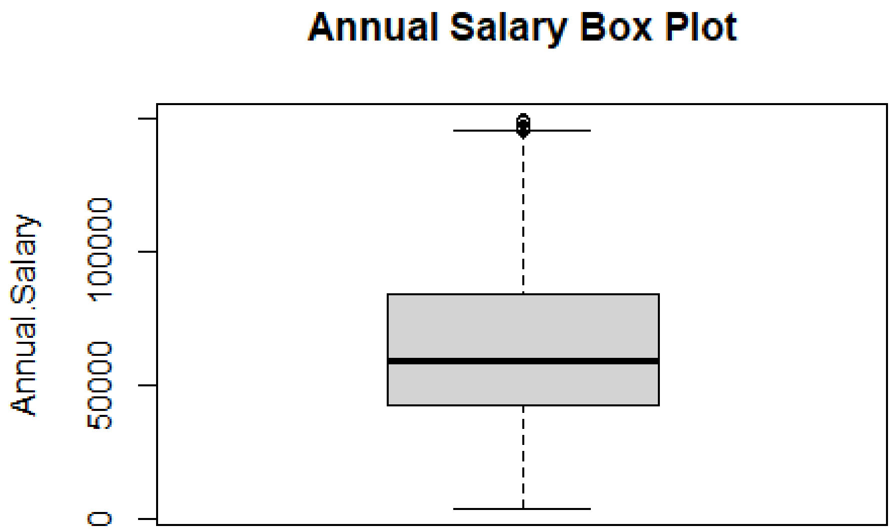
```
### Display "Gross.Year.To.Date" box plot
boxplot(customers$Gross.Year.To.Date,
        main = "Gross Year To Date Box Plot",
        ylab = "Gross.Year.To.Date")
```



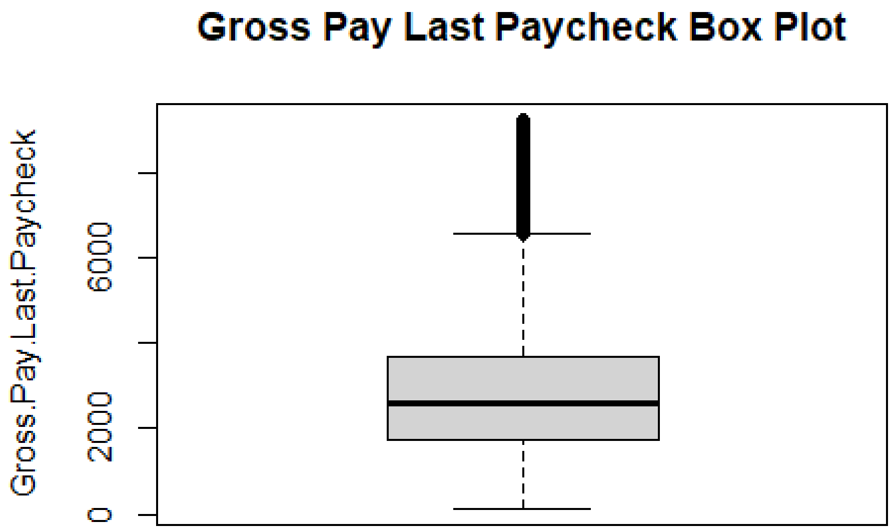
```r
### Display "Gross.Year.To.Date...FRS.Contribution" box plot
boxplot(customers$Gross.Year.To.Date...FRS.Contribution,
        main = "Gross Year To Date ... FRS Contribution Box Plot",
        ylab = "Gross.Year.To.Date...FRS.Contribution")
```



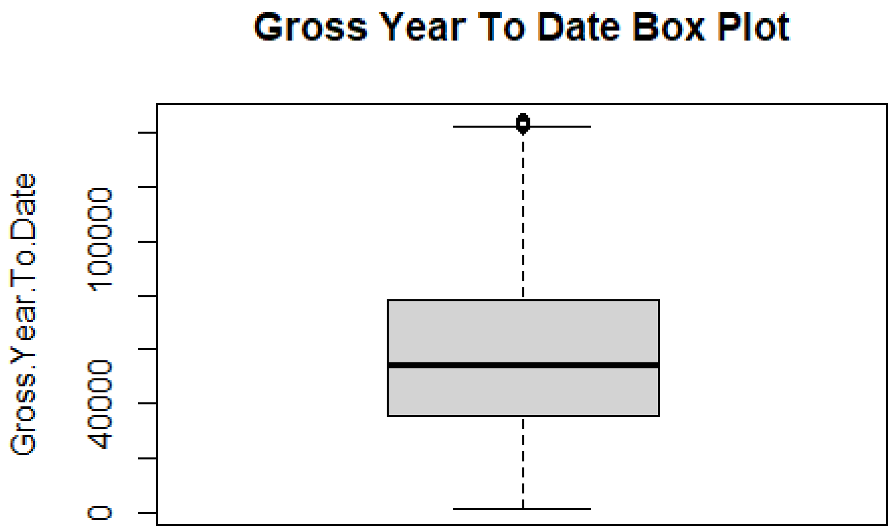
```r
# Capping outliers using the 1st and 99th percentiles
cap_outliers <- function(column) {
  lower_cap <- quantile(column, 0.01)
  upper_cap <- quantile(column, 0.99)
  column[column < lower_cap] <- lower_cap
  column[column > upper_cap] <- upper_cap
  return(column)
}

# Apply capping to the numeric columns
customers$Annual.Salary <- cap_outliers(customers$Annual.Salary)
customers$Gross.Pay.Last.Paycheck <-
cap_outliers(customers$Gross.Pay.Last.Paycheck)
customers$Gross.Year.To.Date <- cap_outliers(customers$Gross.Year.To.Date)
customers$Gross.Year.To.Date...FRS.Contribution <-
cap_outliers(customers$Gross.Year.To.Date...FRS.Contribution)

# Check if outliers are fixed
## Display "Annual.Salary" box plot
boxplot(customers$Annual.Salary,
        main = "Annual Salary Box Plot",
        ylab = "Annual.Salary")
```



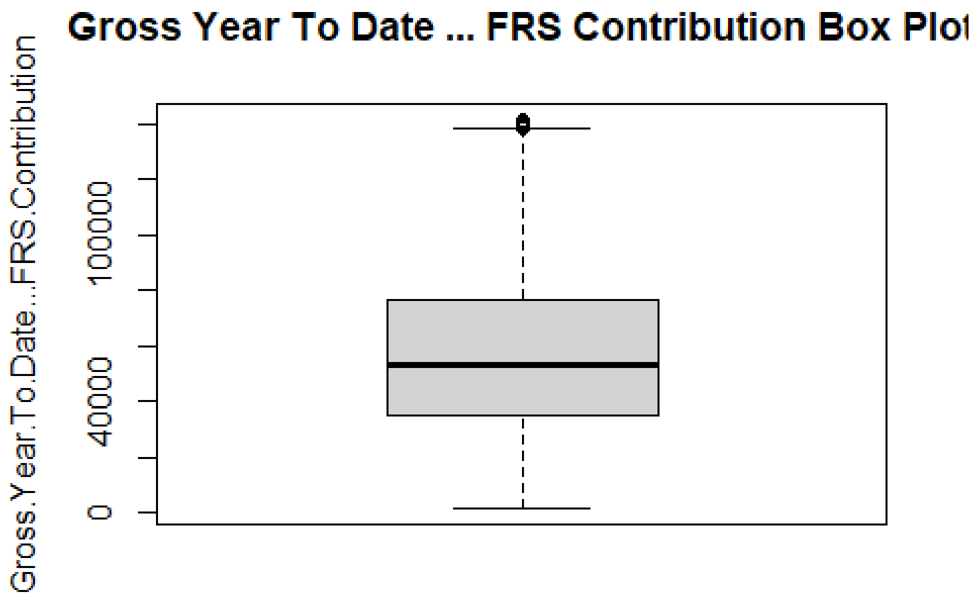
```r
## Display "Gross.Pay.Last.Paycheck" box plot
boxplot(customers$Gross.Pay.Last.Paycheck,
        main = "Gross Pay Last Paycheck Box Plot",
        ylab = "Gross.Pay.Last.Paycheck")
```



```r
## Display "Gross.Year.To.Date" box plot
boxplot(customers$Gross.Year.To.Date,
        main = "Gross Year To Date Box Plot",
        ylab = "Gross.Year.To.Date")
```



```
## Display "Gross.Year.To.Date...FRS.Contribution" box plot
boxplot(customers$Gross.Year.To.Date...FRS.Contribution,
        main = "Gross Year To Date ... FRS Contribution Box Plot",
        ylab = "Gross.Year.To.Date...FRS.Contribution")
```



```
# Check the numerical values
summary(customers)

##     Title           Department.Name    Annual.Salary    
Gross.Pay.Last.Paycheck
##  Length:191317      Length:191317      Min.   :  3744   Min.   : 127.3
##  Class :character   Class :character   1st Qu.: 42537   1st Qu.:1740.1
##  Mode  :character   Mode  :character   Median : 58987   Median :2581.6
##                                        Mean   : 63568   Mean   :2836.2
##                                        3rd Qu.: 83850   3rd Qu.:3682.0
##                                        Max.   :149446   Max.   :9243.5
##  Gross.Year.To.Date Gross.Year.To.Date...FRS.Contribution      Age
##  Min.   :  1540     Min.   :  1511                        Min.   : 34.00
##  1st Qu.: 35984     1st Qu.: 35030                        1st Qu.: 54.00
##  Median : 54703     Median : 53170                        Median : 68.00
##  Mean   : 57662     Mean   : 56124                        Mean   : 66.68
##  3rd Qu.: 78555     3rd Qu.: 76446                        3rd Qu.: 78.00
##  Max.   :144597     Max.   :141468                        Max.   :111.00
##  marital_status       Country_id      Education          Occupation
##  Length:191317      Min.   :52769   Length:191317      Length:191317
##  Class :character   1st Qu.:52776   Class :character   Class :character
##  Mode  :character   Median :52779   Mode  :character   Mode  :character
##                     Mean   :52782
##                     3rd Qu.:52790
##                     Max.   :52791
##  household_size yrs_residence
##  Min.   :2.00   Min.   :2.000
##  1st Qu.:2.00   1st Qu.:2.000  
##  Median :2.00   Median :3.000  
##  Mean   :2.13   Mean   :3.259  
##  3rd Qu.:2.00   3rd Qu.:4.000  
##  Max.   :3.00   Max.   :5.000
```

```
#Assign customers to custData for Aggregation and Tranformation
custData <- customers

#Rename Columns
names(custData)[2] <- 'Department_Name'
names(custData)[3] <- 'Annual_Salary'
names(custData)[4] <- 'Gross_Pay_Last_Paycheck'
names(custData)[5] <- 'Gross_Year_To_Date'
names(custData)[6] <- 'Gross_Year_To_Date_FRS_Contribution'
names(custData)[8] <- 'Marital_Status'
names(custData)[9] <- 'Country_ID'
names(custData)[12] <- 'Household_Size'
names(custData)[13] <- 'Years_Residence'
names(custData)
```

```
##  [1] "Title"                               "Department_Name"                    
##  [3] "Annual_Salary"                       "Gross_Pay_Last_Paycheck"            
##  [5] "Gross_Year_To_Date"                 
"Gross_Year_To_Date_FRS_Contribution"
##  [7] "Age"                                 "Marital_Status"                     
##  [9] "Country_ID"                          "Education"                          
## [11] "Occupation"                          "Household_Size"                     
## [13] "Years_Residence"
```

##DATA AGGREGATION

```
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
```

```
#Sum of Annual Salary by Department Name
Salary_By_Department <- custData %>%
  group_by(Department_Name) %>%
  summarise(Total_Annual_Salary = sum(Annual_Salary))

Salary_By_Department
```

```
## # A tibble: 42 × 2
##    Department_Name                          Total_Annual_Salary
##    <chr>                                                  <dbl>
##  1 ANIMAL SERVICES                                    69312291.
##  2 AUDIT AND MANAGEMENT SERVICES                      20683832.
##  3 AVIATION                                          566935448.
##  4 BOARD OF COUNTY COMMISSIONERS                      73848908.
##  5 CAREERSOURCE SOUTH FLORIDA                         30157891.
##  6 CITIZENS' INDEPENDENT TRANSPORTION TRUST            5756556.
##  7 CLERK OF COURTS                                   365389323 
##  8 COMMISSION ON ETHICS & PUBLIC TRUST                 9630251.
##  9 COMMUNICATIONS DEPARTMENT                          68393706.
## 10 COMMUNITY ACTION AND HUMAN SERVICES               169540282.
## # ℹ 32 more rows
```

```
#Average Annual Pay by Title
Average_Salary_By_Title <- custData %>%
  group_by(Title) %>%
  summarise(Average_Salary = mean(Annual_Salary))

Average_Salary_By_Title
```

```
## # A tibble: 2,290 × 2
##    Title                          Average_Salary
##    <chr>                                   <dbl>
##  1 311 CALL CENTER SPECIALIST             51464.
##  2 311 CALL CENTER SUPERVISOR             75497.
##  3 311 SENIOR CALL CENTER SPCLIST         60267.
##  4 311 SENIOR CALL CENTER SUPV            85350.
##  5 ACCOUNT CLERK                          39538.
##  6 ACCOUNTANT 1                           52101.
##  7 ACCOUNTANT 2                           71368.
##  8 ACCOUNTANT 3                           86149.
##  9 ACCOUNTANT 4                           97388.
## 10 ACCREDITATION MANAGER                  97603.
## # ℹ 2,280 more rows
```

```
#Customers by Eduaction Level
Customers_By_Education <- custData %>%
  group_by(Education) %>%
  summarise(Count = n())

Customers_By_Education
```

```
## # A tibble: 3 × 2
##   Education Count
##   <chr>     <int>
## 1 Bach.     80321
## 2 HS-grad   55498
## 3 Masters   55498
```

```
#Average Gross Year To Date by Age
Gross_Year_By_Age <- custData %>%
  group_by(Age) %>%
  summarise(Average_Gross_Year = mean(Gross_Year_To_Date))

Gross_Year_By_Age
```

```
## # A tibble: 75 × 2
##      Age Average_Gross_Year
##    <int>              <dbl>
##  1    34             54587.
##  2    35             59526.
##  3    36             58215.
##  4    37             57874.
##  5    38             56351.
##  6    39             58885.
##  7    40             57476.
##  8    41             57641.
##  9    42             56926.
## 10    43             57102.
## # ℹ 65 more rows
```

```
# Average Household Size by Years of Residence
Household_Years_Residence <- custData %>%
  group_by(Years_Residence) %>%
  summarise(Average_Household_Size = mean(Household_Size))

Household_Years_Residence
```

```
## # A tibble: 4 × 2
##   Years_Residence Average_Household_Size
##             <int>                  <dbl>
## 1               2                      2
## 2               3                      2
## 3               4                      2
## 4               5                      3
```

```
#Average Annual Salary by Education
Salary_By_Education <- custData %>%
  group_by(Education) %>%
  summarise(Average_Salary_Education = mean(Annual_Salary))

Salary_By_Education
```

```
## # A tibble: 3 × 2
##   Education Average_Salary_Education
##   <chr>                        <dbl>
## 1 Bach.                       63632.
## 2 HS-grad                     63251.
## 3 Masters                     63793.
```

```
#Age by Occupation
Age_By_Occupation <- custData %>%
  group_by(Occupation) %>%
  summarise(Average_Age = mean(Age))

Age_By_Occupation
```

```
## # A tibble: 4 × 2
##   Occupation Average_Age
##   <chr>            <dbl>
## 1 Cleric.           66.6
## 2 Exec.             67.3
## 3 Prof.             66.6
## 4 Sales             66.6
```

```
# Number of Customers by Country
Employees_By_Country <- custData %>%
  group_by(Country_ID) %>%
  summarise(Count = n())

Employees_By_Country
```

```
## # A tibble: 19 × 2
##    Country_ID Count
##         <int> <int>
##  1      52769  2079
##  2      52770 27085
##  3      52771  2488
##  4      52772  6998
##  5      52773  1331
##  6      52774  2862
##  7      52775  2870
##  8      52776 28501
##  9      52777  1316
## 10      52778  7093
## 11      52779 13349
## 12      52782  2163
## 13      52785   837
## 14      52786  2471
## 15      52787   255
## 16      52788   307
## 17      52789 26392
## 18      52790 62623
## 19      52791   297
```

##DATA TRANSFORMATION

```
#Categorisation
length(unique(custData$Title))
```

```
## [1] 2290
```

```
length(unique(custData$Department_Name))
```

```
## [1] 42
```

```
length(unique(custData$Marital_Status))
```

```
## [1] 4
```

```
length(unique(custData$Education))
```

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

```
length(unique(custData$Occupation))
```

```
## [1] 4
```

```
#Categorise Marital Status
custData$Marital_Status <- as.factor(custData$Marital_Status)
table(custData$Marital_Status)
```

```
## 
## divorced  married   single  widowed 
##     2697    55788   132199      633
```

```
#Categorise Education
custData$Education <- as.factor(custData$Education)
table(custData$Education)
```

```
## 
##   Bach. HS-grad Masters 
##   80321   55498   55498
```

```
#Categorise Occupation
custData$Occupation <- as.factor(custData$Occupation)
table(custData$Occupation)
```

```
## 
## Cleric.   Exec.   Prof.   Sales 
##   55498   24823   55498   55498
```

```
#Bin Salary
summary(custData$Annual_Salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    3744   42537   58987   63568   83850  149446
```

```
custData$Salary_Group <- cut(custData$Annual_Salary, breaks = c(0, 42537, 
58987, 83850, Inf),
                             labels = c("Low", "Medium", "High", "Very 
High"))
table(custData$Salary_Group)
```

```
## 
##       Low    Medium      High Very High 
##     47814     47874     46540     49089

#Frequency Encoding:
#Title Encoding:
Title_Frequency <- table(custData$Title)
Title_Frequency_DF <- data.frame(Title = names(Title_Frequency),
Frequency_Title = as.vector(Title_Frequency))
custData <- merge(custData, Title_Frequency_DF, by = "Title")
head(custData$Frequency_Title)

## [1] 517 517 517 517 517 517

#Department Encoding:
Department_Frequency <- table(custData$Department_Name)
Department_Frequency_DF <- data.frame(Department_Name =
names(Department_Frequency), Frequency_Department =
as.vector(Department_Frequency))
custData <- merge(custData, Department_Frequency_DF, by = "Department_Name")
head(custData$Frequency_Department)

## [1] 1602 1602 1602 1602 1602 1602

names(custData)

##  [1] "Department_Name"                     "Title"
##  [3] "Annual_Salary"                       "Gross_Pay_Last_Paycheck"
##  [5] "Gross_Year_To_Date"
"Gross_Year_To_Date_FRS_Contribution"
##  [7] "Age"                                 "Marital_Status"
##  [9] "Country_ID"                          "Education"
## [11] "Occupation"                          "Household_Size"
## [13] "Years_Residence"                     "Salary_Group"
## [15] "Frequency_Title"                     "Frequency_Department"

#Standardisation/Normalisation
library(e1071)
skewness_Annual_Salary <- skewness(custData$Annual_Salary)
print(skewness_Annual_Salary)

## [1] 0.4673479

skewness_Gross_Pay_Last_Paycheck <-
skewness(custData$Gross_Pay_Last_Paycheck)
print(skewness_Gross_Pay_Last_Paycheck)

## [1] 1.154914

skewness_Gross_Year_To_Date <- skewness(custData$Gross_Year_To_Date)
print(skewness_Gross_Year_To_Date)
```

```
## [1] 0.3794214

skewness_Gross_Year_To_Date_FRS_Contribution <-
skewness(custData$Gross_Year_To_Date_FRS_Contribution)
print(skewness_Gross_Year_To_Date_FRS_Contribution)

## [1] 0.3898762

skewness_Age <- skewness(custData$Age)
print(skewness_Age)

## [1] -0.01893976

#Robust Scaling

library(dplyr)
robustScaling <- function(x)
{
  median <- median(x)
  iqr <- IQR(x)
  return((x-median)/iqr)
}

custData <- custData %>%
  mutate(Annual_Salary = robustScaling(Annual_Salary))

custData <- custData %>%
  mutate(Gross_Pay_Last_Paycheck = robustScaling(Gross_Pay_Last_Paycheck))

custData <- custData %>%
  mutate(Gross_Year_To_Date = robustScaling(Gross_Year_To_Date))

custData <- custData %>%
  mutate(Gross_Year_To_Date_FRS_Contribution =
robustScaling(Gross_Year_To_Date_FRS_Contribution))

#Z-Score Normalisation
custData <- custData %>%
  mutate(Age = (Age - mean(Age)) / sd(Age))

#Observe how the dataset has been transformed
head(custData)

##   Department_Name                    Title Annual_Salary
## 1 ANIMAL SERVICES  ASD OUTREACH SPECIALIST    -0.2670866
## 2 ANIMAL SERVICES      ASD CARE SPECIALIST    -0.7246941
## 3 ANIMAL SERVICES   ASD TRANSPORT OPERATOR    -0.7424101
## 4 ANIMAL SERVICES  SENIOR ASST TO DEPT DIR     0.3540869
## 5 ANIMAL SERVICES  ASD OUTREACH SPECIALIST    -0.2670866
## 6 ANIMAL SERVICES ASD TRANSPORT SPECIALIST    -0.3791788
##   Gross_Pay_Last_Paycheck Gross_Year_To_Date
## 1              -0.2860358         -0.3297919
## 2              -0.7317047         -0.7070034
## 3              -0.6410456         -0.6083285
## 4               0.1510127          0.2386836
## 5              -0.2860358         -0.3297919
## 6              -0.4489853         -0.3965425
##   Gross_Year_To_Date_FRS_Contribution        Age Marital_Status Country_ID
## 1                          -0.3307485  0.4878802         single      52770
## 2                          -0.7068393 -1.2461004         single      52789
## 3                          -0.6084586  0.4211886        married      52776
## 4                           0.2360385  0.1544224         single      52771
## 5                          -0.3307485 -0.3791101        married      52789
## 6                          -0.3973024 -1.5795582         single      52790
##   Education Occupation Household_Size Years_Residence Salary_Group
## 1   Masters      Prof.              2               4       Medium
## 2   HS-grad    Cleric.              2               2          Low
## 3     Bach.      Sales              2               3          Low
## 4   HS-grad    Cleric.              2               2         High
## 5   HS-grad    Cleric.              2               2       Medium
## 6   Masters      Prof.              2               4       Medium
##   Frequency_Title Frequency_Department
## 1              22                 1602
## 2             499                 1602
## 3              32                 1602
## 4              13                 1602
## 5              22                 1602
## 6              12                 1602
```

```
str(custData)
```

```
## 'data.frame':    191317 obs. of  16 variables:
##  $ Department_Name                    : chr  "ANIMAL SERVICES" "ANIMAL
SERVICES" "ANIMAL SERVICES" "ANIMAL SERVICES" ...
##  $ Title                              : chr  "ASD OUTREACH SPECIALIST"
"ASD CARE SPECIALIST" "ASD TRANSPORT OPERATOR" "SENIOR ASST TO DEPT DIR" ...
##  $ Annual_Salary                      : num  -0.267 -0.725 -0.742 0.354 -
0.267 ...
##  $ Gross_Pay_Last_Paycheck            : num  -0.286 -0.732 -0.641 0.151 -
0.286 ...
##  $ Gross_Year_To_Date                 : num  -0.33 -0.707 -0.608 0.239 -
0.33 ...
##  $ Gross_Year_To_Date_FRS_Contribution: num  -0.331 -0.707 -0.608 0.236 -
0.331 ...
##  $ Age                                : num  0.488 -1.246 0.421 0.154 -
0.379 ...
##  $ Marital_Status                     : Factor w/ 4 levels
"divorced","married",..: 3 3 2 3 2 3 3 2 3 3 ...
##  $ Country_ID                         : int  52770 52789 52776 52771 52789
52790 52779 52789 52789 52786 ...
```

```
##  $ Education                          : Factor w/ 3 levels "Bach.","HS-
grad",..: 3 2 1 2 2 3 1 1 2 3 ...
##  $ Occupation                         : Factor w/ 4 levels
"Cleric.","Exec.",..: 3 1 4 1 1 3 4 4 1 3 ...
##  $ Household_Size                     : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ Years_Residence                    : int  4 2 3 2 2 4 3 3 2 4 ...
##  $ Salary_Group                       : Factor w/ 4 levels
"Low","Medium",..: 2 1 1 3 2 2 2 1 2 1 ...
##  $ Frequency_Title                    : int  22 499 32 13 22 12 22 32 13
499 ...
##  $ Frequency_Department               : int  1602 1602 1602 1602 1602 1602
1602 1602 1602 1602 ...
```

```
# Export to CSV file
write.csv(custData, "CustData2_Prepared.csv", row.names = FALSE)
```