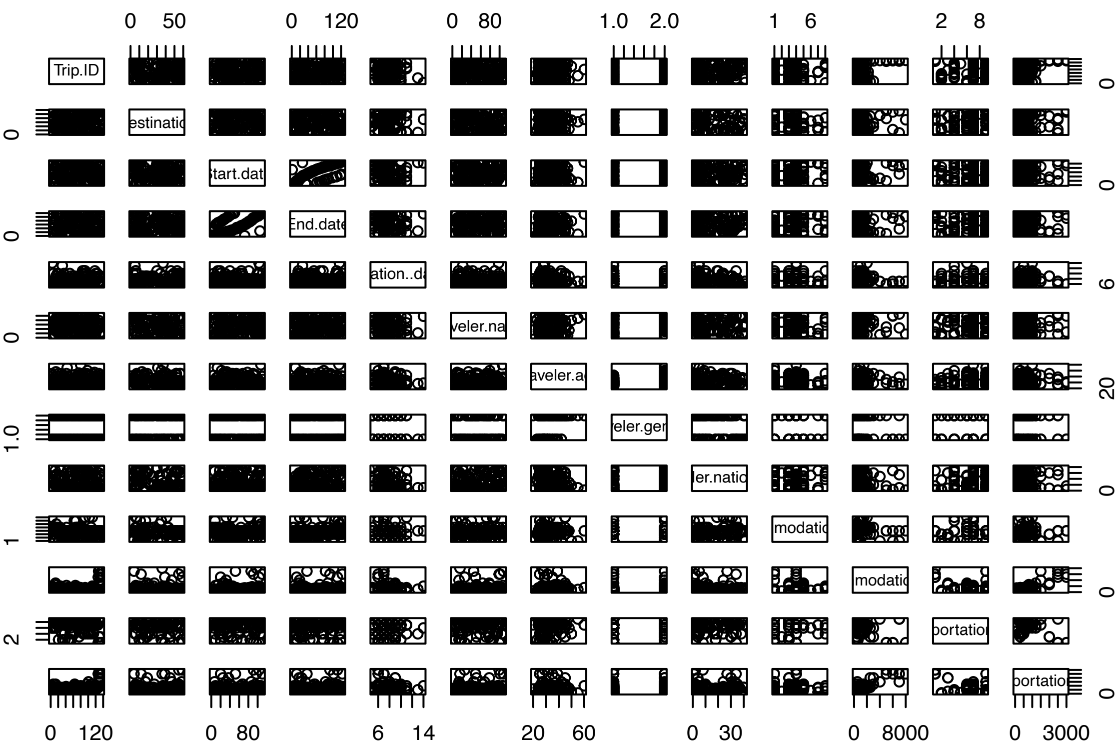
consolidated.R

mappi

2023-03-31

```
## Question 1

# Dataset Source: https://www.kaggle.com/datasets/rkiattisak/traveler-trip-data

library(help = "graphics")
data <- read.csv("Travel details dataset.csv")
###############################################################################################
## (a) Describe the dataset using appropriate plots/curves/charts
plot(data)
```




```
par(mfrow=c(3,3), mar=c(2,5,2,1), las=1, bty="n")
# scatter plot
plot(data$Duration..days., data$Traveler.age, type= "h", xlab = 'Duration Days', ylab = 'Age of Travell
```

```r
# Vertical bar plot
barplot(data$Duration..days., main = 'Duration',xlab = 'Duration', col='red',horiz = FALSE)

# Horizontal bar plot
barplot(data$Traveler.age, main = 'Age',xlab = 'Age', col= 'green',horiz = TRUE)

# Histogram
hist(data$Accommodation.cost, main = 'Accommodation Cost',xlab = 'Cost', col='red')

# Box plots
boxplot(data$Transportation.cost, main = "Transportation Cost")
boxplot(data$Accommodation.cost, main = "Accommodation Cost")

####################################################################################################
## (b) Consider one of continuous attributes, and compute central and variational measures. (8)
age <- data$Traveler.age;
print(age)
```

```
##   [1] 35 28 45 29 26 42 33 25 31 39 27 36 29 48 26 32 30 28 35 45 27 32 29 40 24 34 31 30 45 25 28 33
##  [35] 42 31 27 38 25 33 28 45 30 55 27 41 29 24 31 31 25 27 28 30 23 35 29 27 26 33 35 28 29 43 31 27
##  [69] 33 41 37 35 29 42 46 31 25 38 27 60 32 41 35 28 42 45 31 29 24 26 30 33 27 35 28 45 37 50 31 42
## [103] 29 41 35 28 42 30 26 38 45 31 27 29 33 32 47 26 38 29 41 35 24 30 28 33 35 28 45 31 42 27 37 29
## [137] 39
```

```r
summary(age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.00   28.00   31.00   33.18   38.00   60.00
```

```r
mean_data <- mean(age,na.rm = T)
# mean(age,0.10,na.rm = T)
median_data <- median(age,na.rm = T)

#Mode
calculate_mode <- function(v) {
  uniq_value <- unique(v)
  uniq_value[which.max(tabulate(match(v, uniq_value)))]
}
mode_data <- calculate_mode(age)

print(paste("Central Measures:"))
```

```
## [1] "Central Measures:"
```

```r
print(paste("Mean:", mean_data))
```

```
## [1] "Mean: 33.1751824817518"
```

```r
print(paste("Median:", median_data))
```

```
## [1] "Median: 31"
```

```r
print(paste("Mode:", mode_data))
```

```
## [1] "Mode: 29"
```

```r
# Variational measures
min_data <- min(age)
max_data <- max(age)
age_range <- range(age,na.rm = T)
var_data <- var(age,na.rm = T)
sd_data <- sd(age)
cv_data <- (sd_data/mean_data) * 100
iqr_data <- IQR(age,na.rm = T)
coefficient_Variation <- sd(age)/mean(age)

print(paste("Variational Measures:"))
```

```
## [1] "Variational Measures:"
```

```r
print(paste("Range:", age_range))
```

```
## [1] "Range: 20" "Range: 60"
```

```r
print(paste("Min:", max_data))
```

```
## [1] "Min: 60"
```

```r
print(paste("Max:", min_data))
```

```
## [1] "Max: 20"
```

```r
print(paste("IQR:", iqr_data))
```

```
## [1] "IQR: 10"
```

```r
print(paste("Variance:", var_data))
```

```
## [1] "Variance: 51.0573207385144"
```

```r
print(paste("Standard Deviation:", sd_data))
```

```
## [1] "Standard Deviation: 7.14544055594296"
```

```r
print(paste("Coefficient of Variation:", coefficient_Variation))
```

```
## [1] "Coefficient of Variation: 0.215385116867807"
```

```r
##############################################################################################
## (c) For a particular variable of the dataset, use Chebyshev's rule, and propose one-sigma interval.
# print(data$Traveler.age)
# range(data$Traveler.age)
# diff(range(data$Traveler.age))
# mean(range(data$Traveler.age))
# var(data$Traveler.age)
# sd(data$Traveler.age)
# k <- seq(1:5)
# Cheb <- sapply(k, function(k) 1-1/k^2)
# data.frame(k, Cheb)

traveler_age_mean <- mean(data$Traveler.age)
traveler_age_sd <- sd(data$Traveler.age)

k <- 2 # Can adjust the value of k as needed
prop_within_k_sd <- 1 - 1/k^2
print(prop_within_k_sd)
```

```
## [1] 0.75
```

```r
data.frame(k, prop_within_k_sd)
```

```
##   k prop_within_k_sd
## 1 2             0.75
```

```r
one_sigma_lower <- traveler_age_mean - k * traveler_age_sd
one_sigma_upper <- traveler_age_mean + k * traveler_age_sd


cat("One-sigma interval:", one_sigma_lower, " to ", one_sigma_upper, "\n")
```

```
## One-sigma interval: 18.8843  to  47.46606
```

```r
# Identify outliers as any data points that fall outside the one-sigma interval
outliers <- data$Traveler.age < one_sigma_lower | data$Traveler.age > one_sigma_upper

# Printing total number of outliers
cat("Number of outliers:", sum(outliers), "\n")
```

```
## Number of outliers: 4
```

```r
# Printing the outliers
cat("Outliers list:", which(outliers), "\n")
```

```
## Outliers list: 14 44 80 98
```

```r
##############################################################################################
## (d) Explain how the box-plot technique can be used to detect outliers. Apply this technique for one a

is_outlier <- function(value) {

  # Calculating quartiles, IQR
  Q1 <- quantile(value, probs=.25)
  Q3 <- quantile(value, probs=.75)

  IQR = Q3-Q1

  # Returns true or false based on condition
  return(value > Q3 + (IQR*1.5) | value < Q1 - (IQR*1.5))
}

# Function to get data back with removed outliers
get_removed_outlier_data <- function(data, columns=names(data)) {
  data <- data[!is_outlier(data[[columns[1]]]), ]
  print("Removed outliers successfully!")
  return(data)
}

data <- get_removed_outlier_data(data, c('Transportation.cost'))
```

```
## [1] "Removed outliers successfully!"
```

```r
data <- get_removed_outlier_data(data, c('Accommodation.cost'))
```

```
## [1] "Removed outliers successfully!"
```

```r
# Box plots
boxplot(data$Transportation.cost, main = "Transportation Cost without Outliers")
boxplot(data$Accommodation.cost, main = "Accommodation Cost without Outliers")

##############################################################################################

## Question 2
install.packages("STAT")
```

```
## Error in install.packages : Updating loaded packages
```

```r
install.packages(MASS)
```

```
## Error in install.packages : object 'MASS' not found
```

```r
library(ggplot2)
# Dataset Source: https://www.kaggle.com/datasets/rkiattisak/traveler-trip-data
data <- read.csv("Travel details dataset.csv")

##############################################################################################
```

```r
# a) Select four variables of the dataset, and propose an appropriate probability model to quantify unc

#Normal distribution
age <- data$Traveler.age
age_mean_data <- mean(age,na.rm = T)
age_sd_data <- sd(age)

#Normal distribution: The probability density function: dnorm
norm_pdf <- dnorm(age, mean = age_mean_data, sd = age_sd_data)
norm_data <- data.frame("AGE" = age, "Density" = norm_pdf)
ggplot(norm_data, aes(x = age, y = Density)) + geom_point()

#Normal distribution: The cumulative density function: pnorm
norm_cdf <- pnorm(age, age_mean_data, age_sd_data)
norm_data <- cbind(norm_data, "CDF_LowerTail" = norm_cdf)
cost <- data$Accommodation.cost
ggplot(norm_data, aes(x = age, y = cost)) + geom_point()

#Normal distribution: The quantile function: qnorm
prob.range <- seq(0, 1, 0.001)
qnorm_data <- data.frame("Probability" = prob.range, "Age" = qnorm(prob.range, age_mean_data, age_sd_da
ggplot(qnorm_data, aes(x = Probability, y = Age)) + geom_point()

#Normal distribution: The random sampling function: rnorm
norm_samples <- c(100, 1000, 10000)
norm_dataframe <- do.call(rbind, lapply(norm_samples, function(x) data.frame("samples" = x, "Age" = rno
# show one facet per random sample of a given size
ggplot() + geom_histogram(data = norm_dataframe, aes(x = Age)) + facet_wrap(.~samples, scales = "free_y
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```r
#Bernoulli model
data$Traveler.gender <- ifelse(data$Traveler.gender == "male", 0, 1)
gen_model <- glm(data$Traveler.gender ~ 1, data = data, family = binomial())
```

```
## Warning: glm.fit: algorithm did not converge
```

```r
summary(gen_model)
```

```
##
## Call:
## glm(formula = data$Traveler.gender ~ 1, family = binomial(),
##     data = data)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## 2.409e-06  2.409e-06  2.409e-06  2.409e-06  2.409e-06
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    26.57   30425.66   0.001    0.999
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 0.0000e+00  on 136  degrees of freedom
## Residual deviance: 7.9482e-10  on 136  degrees of freedom
## AIC: 2
##
## Number of Fisher Scoring iterations: 25
```

```r
#Binomial model
Nationality <- data$Traveler.nationality
data$Accommodation.type <- ifelse(data$Accommodation.type == "Hotel", 1, 0)
Accommodation <- data$Accommodation.type
Destination <- data$Destination
Age <- data$Traveler.age
model <- glm(Accommodation ~ Nationality + Age + Destination, data = data, family = binomial())
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(model)
```

```
##
## Call:
## glm(formula = Accommodation ~ Nationality + Age + Destination,
##     family = binomial(), data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -8.49    0.00    0.00    0.00    8.49
##
## Coefficients: (13 not defined because of singularities)
##                                          Estimate Std. Error   z value Pr(>|z|)
## (Intercept)                            -2.341e+14  6.903e+07  -3391433   <2e-16 ***
## NationalityAustralian                   3.992e+15  5.628e+07  70940501   <2e-16 ***
## NationalityBrazil                       5.878e+15  8.933e+07  65798112   <2e-16 ***
## NationalityBrazilian                    1.618e+14  6.760e+07   2393067   <2e-16 ***
## NationalityBritish                     -7.236e+14  3.663e+07 -19754797   <2e-16 ***
## NationalityCambodia                    -3.062e+15  8.933e+07 -34275731   <2e-16 ***
## NationalityCanada                       1.723e+15  7.049e+07  24437930   <2e-16 ***
## NationalityCanadian                     2.379e+15  3.941e+07  60356009   <2e-16 ***
## NationalityChina                        7.715e+14  8.278e+07   9318770   <2e-16 ***
## NationalityChinese                     -1.431e+15  4.783e+07 -29921622   <2e-16 ***
## NationalityDutch                        5.738e+15  7.550e+07  75995558   <2e-16 ***
## NationalityEmirati                     -2.921e+15  9.024e+07 -32373437   <2e-16 ***
## NationalityFrench                       3.378e+15  7.647e+07  44177326   <2e-16 ***
## NationalityGerman                       1.892e+15  9.283e+07  20382343   <2e-16 ***
## NationalityGermany                      6.154e+15  9.069e+07  67856550   <2e-16 ***
## NationalityGreece                      -3.337e+15  8.910e+07 -37451195   <2e-16 ***
## NationalityHong Kong                    9.502e+15  7.971e+07 119205338   <2e-16 ***
## NationalityIndian                      -1.265e+15  7.006e+07 -18054132   <2e-16 ***
## NationalityIndonesian                   2.516e+15  7.851e+07  32049371   <2e-16 ***
## NationalityItalian                     -1.858e+15  6.736e+07 -27589896   <2e-16 ***
## NationalityItaly                        2.415e+15  9.689e+07  24920695   <2e-16 ***
## NationalityJapan                        3.581e+15  9.582e+07  37368886   <2e-16 ***
```

```
## NationalityJapanese                    5.567e+15  7.587e+07  73373207   <2e-16 ***
## NationalityKorean                     -8.082e+13  4.217e+07  -1916542   <2e-16 ***
## NationalityMexican                    -2.936e+14  6.802e+07  -4316652   <2e-16 ***
## NationalityMoroccan                    1.132e+15  8.913e+07  12703655   <2e-16 ***
## NationalityNew Zealander               1.667e+15  9.024e+07  18470232   <2e-16 ***
## NationalityScottish                    5.843e+15  8.924e+07  65474435   <2e-16 ***
## NationalitySingapore                  -2.714e+15  9.181e+07 -29555848   <2e-16 ***
## NationalitySouth African              -2.722e+15  9.181e+07 -29651933   <2e-16 ***
## NationalitySouth Korea                 6.646e+14  5.667e+07  11728117   <2e-16 ***
## NationalitySouth Korean               -1.904e+15  8.535e+07 -22302815   <2e-16 ***
## NationalitySpain                       3.711e+15  1.017e+08  36499525   <2e-16 ***
## NationalitySpanish                    -1.681e+15  7.987e+07 -21047958   <2e-16 ***
## NationalityTaiwan                      1.701e+14  6.591e+07   2580898   <2e-16 ***
## NationalityTaiwanese                   2.252e+15  8.680e+07  25945331   <2e-16 ***
## NationalityUK                          3.953e+15  9.048e+07  43685830   <2e-16 ***
## NationalityUnited Arab Emirates       -2.748e+15  1.318e+08 -20841926   <2e-16 ***
## NationalityUnited Kingdom              3.970e+15  9.064e+07  43805459   <2e-16 ***
## NationalityUSA                         3.240e+15  5.951e+07  54447802   <2e-16 ***
## NationalityVietnamese                 -1.172e+15  5.114e+07 -22916718   <2e-16 ***
## Age                                   -3.454e+13  1.304e+06 -26484097   <2e-16 ***
## DestinationAmsterdam, Netherlands             NA         NA        NA       NA
## DestinationAthens, Greece              1.524e+15  8.586e+07  17749201   <2e-16 ***
## DestinationAuckland, New Zealand              NA         NA        NA       NA
## DestinationAustralia                   6.397e+15  9.275e+07  68973316   <2e-16 ***
## DestinationBali                       -1.159e+15  6.553e+07 -17684425   <2e-16 ***
## DestinationBali, Indonesia            -1.154e+15  6.288e+07 -18356986   <2e-16 ***
## DestinationBangkok                    -1.896e+15  8.753e+07 -21658428   <2e-16 ***
## DestinationBangkok, Thai               5.751e+15  8.507e+07  67603775   <2e-16 ***
## DestinationBangkok, Thailand          -1.955e+15  8.541e+07 -22883179   <2e-16 ***
## DestinationBarcelona                   5.238e+15  1.057e+08  49574392   <2e-16 ***
## DestinationBarcelona, Spain           -1.574e+15  1.005e+08 -15667245   <2e-16 ***
## DestinationBerlin, Germany                    NA         NA        NA       NA
## DestinationBrazil                     -7.122e+15  9.771e+07 -72890156   <2e-16 ***
## DestinationCanada                     -2.445e+15  9.817e+07 -24911172   <2e-16 ***
## DestinationCancun, Mexico              1.719e+15  7.790e+07  22072641   <2e-16 ***
## DestinationCape Town                  -2.544e+15  9.271e+07 -27442016   <2e-16 ***
## DestinationCape Town, SA              -4.690e+14  9.665e+07  -4852801   <2e-16 ***
## DestinationCape Town, South Africa            NA         NA        NA       NA
## DestinationDubai                       8.487e+15  9.690e+07  87584864   <2e-16 ***
## DestinationDubai, United Arab Emirates        NA         NA        NA       NA
## DestinationEdinburgh, Scotland                NA         NA        NA       NA
## DestinationEgypt                       3.624e+15  8.607e+07  42102815   <2e-16 ***
## DestinationFrance                     -3.339e+15  8.507e+07 -39242146   <2e-16 ***
## DestinationGreece                      9.250e+14  8.970e+07  10312367   <2e-16 ***
## DestinationHawaii                      1.489e+15  8.569e+07  17381041   <2e-16 ***
## DestinationHonolulu, Hawaii           -1.809e+15  9.673e+07 -18705088   <2e-16 ***
## DestinationItaly                       1.305e+15  8.918e+07  14633471   <2e-16 ***
## DestinationJapan                       5.774e+15  8.913e+07  64783727   <2e-16 ***
## DestinationLondon                      2.063e+15  7.963e+07  25908902   <2e-16 ***
## DestinationLondon, UK                  1.857e+15  7.289e+07  25478855   <2e-16 ***
## DestinationLos Angeles, USA            7.077e+15  9.667e+07  73207564   <2e-16 ***
## DestinationMarrakech, Morocco                 NA         NA        NA       NA
## DestinationMexico                      2.408e+15  9.383e+07  25662928   <2e-16 ***
## DestinationNew York                   -3.314e+15  7.025e+07 -47173861   <2e-16 ***
```

```
## DestinationNew York City, USA          5.898e+15  7.581e+07  77800539   <2e-16 ***
## DestinationNew York, USA               5.443e+15  6.742e+07  80726357   <2e-16 ***
## DestinationParis                       2.706e+15  6.816e+07  39704492   <2e-16 ***
## DestinationParis, France               3.159e+15  6.029e+07  52395497   <2e-16 ***
## DestinationPhnom Penh                         NA         NA         NA       NA
## DestinationPhuket                             NA         NA         NA       NA
## DestinationPhuket, Thai               -2.423e+15  9.270e+07 -26137753   <2e-16 ***
## DestinationPhuket, Thailand           -1.410e+15  8.507e+07 -16571366   <2e-16 ***
## DestinationRio de Janeiro                     NA         NA         NA       NA
## DestinationRio de Janeiro, Brazil             NA         NA         NA       NA
## DestinationRome                        3.394e+15  8.446e+07  40184132   <2e-16 ***
## DestinationRome, Italy                 9.607e+14  6.492e+07  14798810   <2e-16 ***
## DestinationSantorini                          NA         NA         NA       NA
## DestinationSeoul                       5.751e+15  8.507e+07  67603775   <2e-16 ***
## DestinationSeoul, South Korea          3.057e+15  1.132e+08  27005662   <2e-16 ***
## DestinationSpain                       5.751e+15  8.507e+07  67603774   <2e-16 ***
## DestinationSydney                     -1.545e+15  7.385e+07 -20917386   <2e-16 ***
## DestinationSydney, Aus                 9.389e+14  7.561e+07  12417780   <2e-16 ***
## DestinationSydney, AUS                -1.423e+15  1.132e+08 -12566500   <2e-16 ***
## DestinationSydney, Australia          -2.976e+15  7.141e+07 -41677932   <2e-16 ***
## DestinationThailand                   -5.090e+15  8.827e+07 -57662858   <2e-16 ***
## DestinationTokyo                      -4.072e+15  7.095e+07 -57391909   <2e-16 ***
## DestinationTokyo, Japan                1.057e+15  6.077e+07  17397677   <2e-16 ***
## DestinationVancouver, Canada                  NA         NA         NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance:  187.81  on 136  degrees of freedom
## Residual deviance: 1297.57  on  50  degrees of freedom
## AIC: 1471.6
##
## Number of Fisher Scoring iterations: 23
```

```r
# Multinomial model
# Load the nnet package
library(nnet)
Accommodation <- data$Accommodation.type
Gender <- data$Traveler.gender
Destination <- data$Destination
Age <- data$Traveler.age
dest_model <- multinom(Destination ~ Age + Gender + Accommodation, data = data)
```

```
## # weights:  295 (232 variable)
## initial  value 558.622630
## iter  10 value 522.377509
## iter  20 value 505.690954
## iter  30 value 490.125369
## iter  40 value 477.196263
## iter  50 value 461.183787
## iter  60 value 445.047376
## iter  70 value 433.448886
## iter  80 value 425.758675
```

```
## iter  90 value 423.896479
## iter 100 value 422.905806
## final  value 422.905806
## stopped after 100 iterations
```

```
summary(dest_model)
```

```
## Call:
## multinom(formula = Destination ~ Age + Gender + Accommodation,
##     data = data)
##
## Coefficients:
##                               (Intercept)          Age     Gender Accommodation
## Amsterdam, Netherlands         -1.8903549 -0.195514090 -1.8903549    10.8545464
## Athens, Greece                 -0.3187133  0.078646099 -0.3187133    -8.6944764
## Auckland, New Zealand          -6.3426530  0.115699173 -6.3426530     8.2827056
## Australia                       2.0177693 -0.423679177  2.0177693     9.2042679
## Bali                            1.8466024  0.008252168  1.8466024   -14.7573444
## Bali, Indonesia                 1.4472902  0.016122958  1.4472902    -3.4677055
## Bangkok                         0.2968831  0.083258373  0.2968831   -15.2896059
## Bangkok, Thai                   4.3695609 -0.628828813  4.3695609    10.0806418
## Bangkok, Thailand               4.3012698 -0.182682815  4.3012698   -10.0916434
## Barcelona                       2.8570172 -0.120343978  2.8570172    -1.7438740
## Barcelona, Spain                3.5562955 -0.121530233  3.5562955   -10.6185790
## Berlin, Germany                -9.6952123  0.249636976 -9.6952123     9.2341981
## Brazil                          0.5890593  0.024918382  0.5890593    -8.5105777
## Canada                         -1.5840567  0.146683776 -1.5840567    -9.5896518
## Cancun, Mexico                 -0.7915049  0.124294466 -0.7915049    -2.4577336
## Cape Town                       5.2647763 -0.264756864  5.2647763    -8.5280287
## Cape Town, SA                   1.7826631 -0.048706354  1.7826631    -8.4285510
## Cape Town, South Africa        -3.6507727  0.248251996 -3.6507727   -10.6532545
## Dubai                           5.0631659 -0.696185941  5.0631659    11.1806295
## Dubai, United Arab Emirates    -2.0416317  0.169903243 -2.0416317    -9.6533323
## Edinburgh, Scotland            -1.4830958 -0.152109617 -1.4830958     7.9982583
## Egypt                          -4.9856420  0.059940129 -4.9856420     7.7188092
## France                         14.9361008 -1.032202402 14.9361008    -5.9654016
## Greece                         28.4837892 -2.240186855 28.4837892    -3.6711458
## Hawaii                          0.1017824  0.054195786  0.1017824    -8.5385989
## Honolulu, Hawaii                2.7322260 -0.111961991  2.7322260    -8.6514205
## Italy                           1.7847434 -0.049356079  1.7847434    -8.4445834
## Japan                          -0.1893494 -0.260673396 -0.1893494     8.7426317
## London                          0.9209769  0.005963122  0.9209769    -1.0332185
## London, UK                     -2.2618421  0.180668779 -2.2618421    -1.8510965
## Los Angeles, USA                7.9657375 -0.977326216  7.9657375    12.0134180
## Marrakech, Morocco             10.7550684 -0.701835244 10.7550684    -7.0589686
## Mexico                         -3.0857533  0.220811996 -3.0857533   -10.1227640
## New York                        0.6298027  0.064227431  0.6298027    -3.6684344
## New York City, USA             -2.6286767 -0.069639303 -2.6286767     8.2755231
## New York, USA                  -6.4858255  0.104509027 -6.4858255    10.1157136
## Paris                          -4.3044719 -0.077259199 -4.3044719    13.1305681
## Paris, France                   0.2139152  0.047866098  0.2139152    -0.4472537
## Phnom Penh                      0.5908586  0.025310507  0.5908586    -8.5157518
## Phuket                         -3.6818557  0.248719400 -3.6818557   -10.6687220
## Phuket, Thai                    1.2086721 -0.011004743  1.2086721    -8.6000386
```

```
## Phuket, Thailand             3.8197584 -0.186398499  3.8197584    -8.2010359
## Rio de Janeiro              -2.1992191 -0.102639129 -2.1992191     7.8124013
## Rio de Janeiro, Brazil       3.1224144 -0.101922264  3.1224144    -2.8758157
## Rome                         2.2653933 -0.058749631  2.2653933    -1.4814015
## Rome, Italy                 -0.7341096  0.121210678 -0.7341096    -2.4434470
## Santorini                    6.6829343 -0.393793107  6.6829343    -7.3804888
## Seoul                        4.3408966 -0.628008185  4.3408966    10.0768205
## Seoul, South Korea           6.6915528 -0.395130798  6.6915528    -7.3934990
## Spain                        4.3958943 -0.632121771  4.3958943    10.1087517
## Sydney                      -0.5365363  0.110302926 -0.5365363    -2.0094860
## Sydney, Aus                  2.4579120 -0.070798033  2.4579120    -2.5657980
## Sydney, AUS                  6.6952287 -0.395063018  6.6952287    -7.3904499
## Sydney, Australia            1.8442412 -0.030916978  1.8442412    -2.6943984
## Thailand                    -3.7265201  0.250721486 -3.7265201   -10.6202568
## Tokyo                        2.3867386 -0.043692653  2.3867386    -3.3426429
## Tokyo, Japan                 7.6411896 -0.413559400  7.6411896    -1.3265019
## Vancouver, Canada            9.6102206 -0.615058305  9.6102206    -0.7655682
##
## Std. Errors:
##                             (Intercept)       Age    Gender Accommodation
## Amsterdam, Netherlands         2.750999 0.2467763  2.750999    2.76217289
## Athens, Greece                 4.055752 0.2262074  4.055752   24.46832492
## Auckland, New Zealand          2.855524 0.2270121  2.855524    2.88245150
## Australia                      3.779785 0.3758745  3.779785    3.82365290
## Bali                           3.260891 0.1838645  3.260891    0.04416941
## Bali, Indonesia                3.294799 0.1856281  3.294799    3.16055965
## Bangkok                        3.349162 0.1867353  3.349162    0.01820129
## Bangkok, Thai                  4.362539 0.4582202  4.362539    4.38308142
## Bangkok, Thailand              4.020989 0.2484559  4.020989   41.99754772
## Barcelona                      3.937604 0.2377285  3.937604    3.31913114
## Barcelona, Spain               3.643870 0.2163570  3.643870   43.53468460
## Berlin, Germany                2.809054 0.2093520  2.809054    2.81672679
## Brazil                         4.158093 0.2394181  4.158093   24.38042972
## Canada                         4.132230 0.2217148  4.132230   35.32841226
## Cancun, Mexico                 3.330511 0.1830981  3.330511    3.13987283
## Cape Town                      4.771108 0.3115895  4.771108   25.90844332
## Cape Town, SA                  4.419356 0.2664316  4.419356   25.77529807
## Cape Town, South Africa        4.495467 0.2271147  4.495467   50.48562960
## Dubai                          3.657925 0.3780330  3.657925    3.67337200
## Dubai, United Arab Emirates    4.199179 0.2222259  4.199179   35.56156121
## Edinburgh, Scotland            3.172878 0.2872799  3.172878    3.25457332
## Egypt                          2.892720 0.2368158  2.892720    2.94865602
## France                         8.719658 0.6722300  8.719658   15.16540561
## Greece                        15.835845 1.3850352 15.835845   16.32936011
## Hawaii                         4.087321 0.2312081  4.087321   23.54929159
## Honolulu, Hawaii               4.831638 0.3033012  4.831638   31.86079796
## Italy                          4.441054 0.2680002  4.441054   26.21082624
## Japan                          3.504707 0.3313133  3.504707    3.55107226
## London                         3.366282 0.1888810  3.366282    3.18445499
## London, UK                     3.480077 0.1867029  3.480077    3.24008935
## Los Angeles, USA               5.443980 0.6128856  5.443980    5.44799610
## Marrakech, Morocco             7.887726 0.5851045  7.887726   23.77679617
## Mexico                         4.403508 0.2258563  4.403508   41.80148022
## New York                       3.282565 0.1833416  3.282565    3.17413443
```

```
## New York City, USA            2.535515 0.2202004  2.535515    2.64907897
## New York, USA                 2.324323 0.1901278  2.324323    2.34021803
## Paris                         2.176345 0.1840821  2.176345    2.17976205
## Paris, France                 3.227416 0.1781438  3.227416    3.15634775
## Phnom Penh                    4.141590 0.2383458  4.141590   24.22933445
## Phuket                        4.545066 0.2290587  4.545066   51.89538626
## Phuket, Thai                  4.208070 0.2473857  4.208070   25.98622673
## Phuket, Thailand              5.363326 0.3504266  5.363326   28.10088075
## Rio de Janeiro                3.080818 0.2729335  3.080818    3.16524890
## Rio de Janeiro, Brazil        3.525483 0.2064832  3.525483    3.19922438
## Rome                          3.336404 0.1903505  3.336404    3.10985393
## Rome, Italy                   3.329029 0.1831317  3.329029    3.13885111
## Santorini                     6.767156 0.4752850  6.767156   23.56004769
## Seoul                         4.421091 0.4649682  4.421091    4.44084540
## Seoul, South Korea            6.823992 0.4797779  6.823992   23.95926476
## Spain                         4.382746 0.4608189  4.382746    4.40260137
## Sydney                        3.273531 0.1804154  3.273531    3.10972341
## Sydney, Aus                   3.582219 0.2091070  3.582219    3.22642729
## Sydney, AUS                   6.799453 0.4778788  6.799453   23.81418509
## Sydney, Australia             3.513717 0.2020995  3.513717    3.22320582
## Thailand                      4.558733 0.2294280  4.558733   50.49209751
## Tokyo                         3.359072 0.1920387  3.359072    3.17701208
## Tokyo, Japan                  3.724897 0.2313022  3.724897    3.08426342
## Vancouver, Canada             5.336378 0.3731855  5.336378    3.33483149
##
## Residual Deviance: 845.8116
## AIC: 1193.812
```

```r
# Poisson model
Cost <- data$Transportation.cost
cost_model <- glm(Cost ~ Age + Destination, data = data, family = poisson())
summary(cost_model)
```

```
##
## Call:
## glm(formula = Cost ~ Age + Destination, family = poisson(), data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -46.269   -9.582    0.000    3.659   45.314
##
## Coefficients:
##                                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        4.9890498  0.0853897  58.427  < 2e-16 ***
## Age                                0.0005265  0.0006096   0.864  0.38780
## DestinationAmsterdam, Netherlands  1.1038770  0.0884022  12.487  < 2e-16 ***
## DestinationAthens, Greece          1.7949183  0.0882675  20.335  < 2e-16 ***
## DestinationAuckland, New Zealand   2.8144637  0.0840723  33.477  < 2e-16 ***
## DestinationAustralia               1.2108170  0.0934317  12.959  < 2e-16 ***
## DestinationBali                    1.7307465  0.0828510  20.890  < 2e-16 ***
## DestinationBali, Indonesia         1.6266780  0.0833736  19.511  < 2e-16 ***
## DestinationBangkok                 0.8503196  0.0859834   9.889  < 2e-16 ***
## DestinationBangkok, Thai           0.9881999  0.0961224  10.281  < 2e-16 ***
## DestinationBangkok, Thailand       0.9003083  0.0873612  10.306  < 2e-16 ***
```

```
## DestinationBarcelona                    1.0467186  0.0887711  11.791  < 2e-16 ***
## DestinationBarcelona, Spain             1.7553836  0.0837123  20.969  < 2e-16 ***
## DestinationBerlin, Germany              1.5367597  0.0900747  17.061  < 2e-16 ***
## DestinationBrazil                      -1.0944005  0.1633721  -6.699 2.10e-11 ***
## DestinationCanada                      -0.6915678  0.1414332  -4.890 1.01e-06 ***
## DestinationCancun, Mexico               1.3874723  0.0841735  16.483  < 2e-16 ***
## DestinationCape Town                    1.7414451  0.0855436  20.357  < 2e-16 ***
## DestinationCape Town, SA                0.6984119  0.1001856   6.971 3.14e-12 ***
## DestinationCape Town, South Africa      2.5881613  0.0846913  30.560  < 2e-16 ***
## DestinationDubai                        1.3069160  0.0874863  14.939  < 2e-16 ***
## DestinationDubai, United Arab Emirates  1.6750294  0.0889840  18.824  < 2e-16 ***
## DestinationEdinburgh, Scotland          0.0047383  0.1156003   0.041  0.96731
## DestinationEgypt                       -0.4033592  0.1291225  -3.124  0.00179 **
## DestinationFrance                       0.7015708  0.1004746   6.983 2.90e-12 ***
## DestinationGreece                       1.3973503  0.0921804  15.159  < 2e-16 ***
## DestinationHawaii                       1.6776618  0.0890779  18.834  < 2e-16 ***
## DestinationHonolulu, Hawaii             2.0857592  0.0869110  23.999  < 2e-16 ***
## DestinationItaly                       -0.2178788  0.1226261  -1.777  0.07561 .
## DestinationJapan                        1.2097640  0.0933361  12.961  < 2e-16 ***
## DestinationLondon                       0.8505866  0.0859967   9.891  < 2e-16 ***
## DestinationLondon, UK                   0.6927839  0.0881928   7.855 3.99e-15 ***
## DestinationLos Angeles, USA             0.7010443  0.1004172   6.981 2.92e-12 ***
## DestinationMarrakech, Morocco           0.9887264  0.0961784  10.280  < 2e-16 ***
## DestinationMexico                       1.6734500  0.0889777  18.808  < 2e-16 ***
## DestinationNew York                     0.7608320  0.0854683   8.902  < 2e-16 ***
## DestinationNew York City, USA           2.3062636  0.0837740  27.530  < 2e-16 ***
## DestinationNew York, USA                1.8291494  0.0838160  21.823  < 2e-16 ***
## DestinationParis                        1.2362023  0.0834656  14.811  < 2e-16 ***
## DestinationParis, France                1.7619713  0.0826967  21.306  < 2e-16 ***
## DestinationPhnom Penh                   1.2081846  0.0932226  12.960  < 2e-16 ***
## DestinationPhuket                       1.5383391  0.0900066  17.091  < 2e-16 ***
## DestinationPhuket, Thai                 1.6787147  0.0891447  18.831  < 2e-16 ***
## DestinationPhuket, Thailand             1.2108170  0.0934317  12.959  < 2e-16 ***
## DestinationRio de Janeiro               0.9850411  0.0958668  10.275  < 2e-16 ***
## DestinationRio de Janeiro, Brazil       1.7547253  0.0836506  20.977  < 2e-16 ***
## DestinationRome                         0.8110009  0.0853776   9.499  < 2e-16 ***
## DestinationRome, Italy                  1.5417595  0.0838220  18.393  < 2e-16 ***
## DestinationSantorini                    0.0073706  0.1157850   0.064  0.94924
## DestinationSeoul                        0.0073706  0.1157850   0.064  0.94924
## DestinationSeoul, South Korea          -2.0075324  0.2382006  -8.428  < 2e-16 ***
## DestinationSpain                        0.9881999  0.0961224  10.281  < 2e-16 ***
## DestinationSydney                       1.7932252  0.0830165  21.601  < 2e-16 ***
## DestinationSydney, Aus                  1.7591049  0.0841779  20.897  < 2e-16 ***
## DestinationSydney, AUS                  0.2950527  0.1083490   2.723  0.00647 **
## DestinationSydney, Australia            2.2732462  0.0832134  27.318  < 2e-16 ***
## DestinationThailand                    -0.0021059  0.1154958  -0.018  0.98545
## DestinationTokyo                        1.0346672  0.0847168  12.213  < 2e-16 ***
## DestinationTokyo, Japan                 1.4169690  0.0834724  16.975  < 2e-16 ***
## DestinationVancouver, Canada            2.3590083  0.0840370  28.071  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
```

```
##     Null deviance: 59312  on 136  degrees of freedom
## Residual deviance: 31400  on  77  degrees of freedom
## AIC: 32605
##
## Number of Fisher Scoring iterations: 5
```

```r
# Geometric model to the Number of Previous Visits variable
library(STAT)
library(MASS)
Accommodation <- data$Accommodation.type
Gender <- data$Traveler.gender
Destination <- data$Destination
Age <- data$Traveler.age
# Binomial distribution with a logit link function, which is equivalent to a geometric distribution.
acco_model <- glm(Accommodation ~ Age + Gender + Destination, data = data, family = binomial(link = "lo
summary(acco_model)
```

```
##
## Call:
## glm(formula = Accommodation ~ Age + Gender + Destination, family = binomial(link = "logit"),
##     data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.96902  -0.47317  -0.00007   0.00010   1.84059
##
## Coefficients: (1 not defined because of singularities)
##                                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)                              1.650e+01  1.075e+04   0.002    0.999
## Age                                      7.466e-02  4.806e-02   1.554    0.120
## Gender                                          NA         NA      NA       NA
## DestinationAmsterdam, Netherlands        7.466e-01  1.317e+04   0.000    1.000
## DestinationAthens, Greece               -3.868e+01  1.521e+04  -0.003    0.998
## DestinationAuckland, New Zealand         1.493e-01  1.521e+04   0.000    1.000
## DestinationAustralia                     9.706e-01  1.521e+04   0.000    1.000
## DestinationBali                         -3.853e+01  1.148e+04  -0.003    0.997
## DestinationBali, Indonesia              -2.046e+01  1.075e+04  -0.002    0.998
## DestinationBangkok                      -3.877e+01  1.198e+04  -0.003    0.997
## DestinationBangkok, Thai                 1.045e+00  1.521e+04   0.000    1.000
## DestinationBangkok, Thailand            -3.825e+01  1.239e+04  -0.003    0.998
## DestinationBarcelona                    -1.882e+01  1.075e+04  -0.002    0.999
## DestinationBarcelona, Spain             -3.829e+01  1.200e+04  -0.003    0.997
## DestinationBerlin, Germany              -5.226e-01  1.521e+04   0.000    1.000
## DestinationBrazil                       -3.853e+01  1.521e+04  -0.003    0.998
## DestinationCanada                       -3.891e+01  1.521e+04  -0.003    0.998
## DestinationCancun, Mexico               -1.941e+01  1.075e+04  -0.002    0.999
## DestinationCape Town                    -3.816e+01  1.317e+04  -0.003    0.998
## DestinationCape Town, SA                -3.839e+01  1.521e+04  -0.003    0.998
## DestinationCape Town, South Africa      -3.943e+01  1.521e+04  -0.003    0.998
## DestinationDubai                         1.094e+00  1.316e+04   0.000    1.000
## DestinationDubai, United Arab Emirates -3.898e+01  1.521e+04  -0.003    0.998
## DestinationEdinburgh, Scotland           6.720e-01  1.521e+04   0.000    1.000
## DestinationEgypt                         2.987e-01  1.521e+04   0.000    1.000
## DestinationFrance                       -3.794e+01  1.521e+04  -0.002    0.998
```

```
## DestinationGreece                        -3.756e+01  1.521e+04  -0.002    0.998
## DestinationHawaii                        -3.861e+01  1.521e+04  -0.003    0.998
## DestinationHonolulu, Hawaii              -3.824e+01  1.521e+04  -0.003    0.998
## DestinationItaly                         -3.839e+01  1.521e+04  -0.003    0.998
## DestinationJapan                          8.213e-01  1.521e+04   0.000    1.000
## DestinationLondon                        -1.798e+01  1.075e+04  -0.002    0.999
## DestinationLondon, UK                    -1.883e+01  1.075e+04  -0.002    0.999
## DestinationLos Angeles, USA               1.120e+00  1.521e+04   0.000    1.000
## DestinationMarrakech, Morocco            -3.801e+01  1.521e+04  -0.002    0.998
## DestinationMexico                        -3.921e+01  1.521e+04  -0.003    0.998
## DestinationNew York                      -2.065e+01  1.075e+04  -0.002    0.998
## DestinationNew York City, USA             6.082e-01  1.307e+04   0.000    1.000
## DestinationNew York, USA                  4.269e-01  1.220e+04   0.000    1.000
## DestinationParis                          6.170e-01  1.148e+04   0.000    1.000
## DestinationParis, France                 -1.729e+01  1.075e+04  -0.002    0.999
## DestinationPhnom Penh                    -3.853e+01  1.521e+04  -0.003    0.998
## DestinationPhuket                        -3.943e+01  1.521e+04  -0.003    0.998
## DestinationPhuket, Thai                  -3.846e+01  1.521e+04  -0.003    0.998
## DestinationPhuket, Thailand              -3.816e+01  1.521e+04  -0.003    0.998
## DestinationRio de Janeiro                 5.973e-01  1.521e+04   0.000    1.000
## DestinationRio de Janeiro, Brazil        -1.991e+01  1.075e+04  -0.002    0.999
## DestinationRome                          -1.850e+01  1.075e+04  -0.002    0.999
## DestinationRome, Italy                   -1.938e+01  1.075e+04  -0.002    0.999
## DestinationSantorini                     -3.809e+01  1.521e+04  -0.003    0.998
## DestinationSeoul                          1.045e+00  1.521e+04   0.000    1.000
## DestinationSeoul, South Korea            -3.809e+01  1.521e+04  -0.003    0.998
## DestinationSpain                          1.045e+00  1.521e+04   0.000    1.000
## DestinationSydney                        -1.893e+01  1.075e+04  -0.002    0.999
## DestinationSydney, Aus                   -1.956e+01  1.075e+04  -0.002    0.999
## DestinationSydney, AUS                   -3.809e+01  1.521e+04  -0.003    0.998
## DestinationSydney, Australia             -1.961e+01  1.075e+04  -0.002    0.999
## DestinationThailand                      -3.943e+01  1.521e+04  -0.003    0.998
## DestinationTokyo                         -2.031e+01  1.075e+04  -0.002    0.998
## DestinationTokyo, Japan                  -1.884e+01  1.075e+04  -0.002    0.999
## DestinationVancouver, Canada             -1.848e+01  1.075e+04  -0.002    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 187.807  on 136  degrees of freedom
## Residual deviance:  78.315  on  77  degrees of freedom
## AIC: 198.32
##
## Number of Fisher Scoring iterations: 18
```

```
##################################################################################################

# b) For each model in part (a), estimate the parameters of model.

#Normal distribution:
days <- data$Duration..days

days_mu <- mean(days)
days_sigma <- sd(days)
```

```r
cat("Estimated mean:", days_mu, "\n")
```

```
## Estimated mean: 7.605839
```

```r
cat("Estimated standard deviation:", days_sigma, "\n")
```

```
## Estimated standard deviation: 1.601276
```

```r
#Bernoulli model
data$Duration..days <- ifelse(data$Duration..days > 10, 5, 0)
days <- data$Duration..days
days_prob <- mean(days)
cat("Estimated probability of a long stay:", days_prob, "\n")
```

```
## Estimated probability of a long stay: 0.2554745
```

```r
#Binomial model
data$Duration..days <- ifelse(data$Duration..days > 10, 5, 0)
days <- data$Duration..days
Trips <- 1
model <- glm(cbind(days, Trips - days) ~ Destination, data = data, family = "binomial")
```

```
## Warning: glm.fit: algorithm did not converge
```

```r
summary(model)
```

```
##
## Call:
## glm(formula = cbind(days, Trips - days) ~ Destination, family = "binomial",
##     data = data)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.409e-06  -2.409e-06  -2.409e-06  -2.409e-06  -2.409e-06
##
## Coefficients:
##                                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           -2.657e+01  3.561e+05       0        1
## DestinationAmsterdam, Netherlands     -2.381e-28  4.362e+05       0        1
## DestinationAthens, Greece              2.999e-28  5.036e+05       0        1
## DestinationAuckland, New Zealand      -5.784e-28  5.036e+05       0        1
## DestinationAustralia                  -1.711e-28  5.036e+05       0        1
## DestinationBali                       -1.644e-28  3.807e+05       0        1
## DestinationBali, Indonesia            -3.410e-28  3.901e+05       0        1
## DestinationBangkok                    -1.599e-28  3.982e+05       0        1
## DestinationBangkok, Thai               8.331e-29  5.036e+05       0        1
## DestinationBangkok, Thailand          -1.632e-28  4.112e+05       0        1
## DestinationBarcelona                  -1.655e-28  4.362e+05       0        1
## DestinationBarcelona, Spain           -2.247e-28  3.982e+05       0        1
## DestinationBerlin, Germany            -2.527e-28  5.036e+05       0        1
```

```
## DestinationBrazil                        -2.316e-28  5.036e+05       0        1
## DestinationCanada                        -1.829e-28  5.036e+05       0        1
## DestinationCancun, Mexico                -2.595e-28  3.982e+05       0        1
## DestinationCape Town                     -2.697e-28  4.362e+05       0        1
## DestinationCape Town, SA                 -1.043e-28  5.036e+05       0        1
## DestinationCape Town, South Africa       -1.099e-28  5.036e+05       0        1
## DestinationDubai                         -2.693e-28  4.362e+05       0        1
## DestinationDubai, United Arab Emirates -2.042e-28  5.036e+05       0        1
## DestinationEdinburgh, Scotland           -3.302e-28  5.036e+05       0        1
## DestinationEgypt                         -2.330e-28  5.036e+05       0        1
## DestinationFrance                        -2.925e-28  5.036e+05       0        1
## DestinationGreece                        -2.169e-28  5.036e+05       0        1
## DestinationHawaii                        -3.989e-29  5.036e+05       0        1
## DestinationHonolulu, Hawaii              -1.888e-28  5.036e+05       0        1
## DestinationItaly                         -2.080e-28  5.036e+05       0        1
## DestinationJapan                         -2.078e-28  5.036e+05       0        1
## DestinationLondon                        -1.537e-28  3.982e+05       0        1
## DestinationLondon, UK                    -5.043e-13  4.112e+05       0        1
## DestinationLos Angeles, USA              -5.423e-29  5.036e+05       0        1
## DestinationMarrakech, Morocco            -1.275e-28  5.036e+05       0        1
## DestinationMexico                        -1.275e-28  5.036e+05       0        1
## DestinationNew York                      -1.565e-28  3.901e+05       0        1
## DestinationNew York City, USA            -2.760e-28  4.362e+05       0        1
## DestinationNew York, USA                  1.452e-29  4.112e+05       0        1
## DestinationParis                         -1.253e-28  3.807e+05       0        1
## DestinationParis, France                 -1.985e-28  3.807e+05       0        1
## DestinationPhnom Penh                    -2.278e-28  5.036e+05       0        1
## DestinationPhuket                        -2.169e-28  5.036e+05       0        1
## DestinationPhuket, Thai                  -1.447e-28  5.036e+05       0        1
## DestinationPhuket, Thailand              -4.089e-28  5.036e+05       0        1
## DestinationRio de Janeiro                 2.062e-28  5.036e+05       0        1
## DestinationRio de Janeiro, Brazil        -1.918e-28  3.982e+05       0        1
## DestinationRome                          -2.376e-28  3.901e+05       0        1
## DestinationRome, Italy                   -2.154e-28  3.982e+05       0        1
## DestinationSantorini                     -1.958e-28  5.036e+05       0        1
## DestinationSeoul                         -2.252e-28  5.036e+05       0        1
## DestinationSeoul, South Korea            -2.131e-28  5.036e+05       0        1
## DestinationSpain                         -1.864e-28  5.036e+05       0        1
## DestinationSydney                        -1.532e-28  3.901e+05       0        1
## DestinationSydney, Aus                   -1.680e-28  4.112e+05       0        1
## DestinationSydney, AUS                   -2.184e-28  5.036e+05       0        1
## DestinationSydney, Australia             -1.308e-28  4.112e+05       0        1
## DestinationThailand                      -1.189e-28  5.036e+05       0        1
## DestinationTokyo                         -1.625e-28  3.901e+05       0        1
## DestinationTokyo, Japan                   2.888e-28  3.807e+05       0        1
## DestinationVancouver, Canada             -1.680e-28  4.362e+05       0        1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 0.0000e+00  on 136  degrees of freedom
## Residual deviance: 7.9482e-10  on  78  degrees of freedom
## AIC: 118
##
## Number of Fisher Scoring iterations: 25
```

```
# Multinomial model
# Load the nnet package
library(nnet)
Accommodation <- data$Accommodation.type
# Define the outcome variable as a factor with three levels
StayLength <- cut(jitter(data$Duration..days), breaks = quantile(jitter(data$Duration..days), probs = c
print(StayLength)
```

```
##   [1] long   long   short  short  medium short  short  long   long   medium long   long   short  medi
##  [15] short  medium long   medium long   medium short  medium short  medium medium long   short  medi
##  [29] short  long   short  short  <NA>   long   medium short  medium short  short  short  medium sho
##  [43] medium medium long   medium short  short  medium long   medium long   medium long   medium medi
##  [57] medium medium medium medium medium short  long   long   long   medium long   long   medium lon
##  [71] long   long   long   long   long   medium medium long   medium short  long   medium short  lon
##  [85] short  medium short  short  long   <NA>   long   long   short  short  medium long   long   lon
##  [99] long   short  long   short  long   short  medium short  medium medium medium medium medium medi
## [113] medium medium long   short  long   long   short  medium long   medium short  medium long   sho
## [127] medium medium long   short  medium short  short  short  medium medium long
## Levels: short medium long
```

```
dest_model <- multinom(StayLength ~ Accommodation, data = data)
```

```
## # weights:  9 (4 variable)
## initial  value 148.312659
## final  value 147.437676
## converged
```

```
summary(dest_model)
```

```
## Call:
## multinom(formula = StayLength ~ Accommodation, data = data)
##
## Coefficients:
##        (Intercept) Accommodation
## medium   0.2876821   -0.04255982
## long     0.2135744   -0.15950717
##
## Std. Errors:
##        (Intercept) Accommodation
## medium   0.2886751     0.4270447
## long     0.2933949     0.4407581
##
## Residual Deviance: 294.8754
## AIC: 302.8754
```

```
# Poisson model
Cost <- data$Transportation.cost
data$Traveler.gender <- ifelse(data$Traveler.gender == "male", 0, 1)
Gender <- data$Traveler.gender
poi_model <- glm(Gender ~ Cost, data = data, family = poisson)
summary(poi_model)
```

```
##
## Call:
## glm(formula = Gender ~ Cost, family = poisson, data = data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##      0       0       0       0       0
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.291e-17  1.279e-01       0        1
## Cost        -5.611e-37  1.472e-04       0        1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 0  on 136  degrees of freedom
## Residual deviance: 0  on 135  degrees of freedom
## AIC: 278
##
## Number of Fisher Scoring iterations: 4
```

```
intercept <- coef(poi_model)[1]
avg_rate <- exp(intercept)
cat("Gender who travels most", round(avg_rate, 2))
```

```
## Gender who travels most 1
```

```
####################################################################################################

# c) Express the way in which each model can be used for the predictive analytics, then find the predic

#Bernoulli model
#Example: Suppose we want to predict the gender of a traveler based on their age, nationality, and dest
Gender <- ifelse(data$Traveler.gender == "male", 0, 1)
Destination <- data$Destination
Age <- data$Traveler.age
Nationality <- data$Traveler.nationality
gender_model <- glm(Gender ~ Age + Nationality + Destination, data = data, family = binomial())
```

```
## Warning: glm.fit: algorithm did not converge
```

```
gender_prediction <- predict(gender_model, type = "response")
print(gender_prediction)
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
##  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
```

```
## 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1 
## 131 132 133 134 135 136 137
##   1   1   1   1   1   1   1
```

```r
# Binomial model:
# Example: To predict the accommodation type for each traveler in the dataset based on their nationalit
Nationality <- data$Traveler.nationality
Age <- data$Traveler.age
Destination <- data$Destination
Accommodation <- data$Accommodation.type
accommodation_model <- glm(Accommodation ~ Nationality + Age + Destination, data = data, family = binom
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
accommodation_prediction <- predict(accommodation_model, type = "response")
print(accommodation_prediction)
```

```
##            1            2            3            4            5            6            7 
## 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e
##            9           10           11           12           13           14           15 
## 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e
##           17           18           19           20           21           22           23 
## 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 1.000000e+00 1.000000e+00 2.220446e
##           25           26           27           28           29           30           31 
## 1.000000e+00 2.220446e-16 1.000000e+00 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e
##           33           34           35           36           37           38           39 
## 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e
##           41           42           43           44           45           46           47 
## 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e
##           49           50           51           52           53           54           55 
## 1.000000e+00 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e
##           57           58           59           60           61           62           63 
## 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e+00 1.000000e+00 2.220446e-16 1.000000e
##           65           66           67           68           69           70           71 
## 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e-16 2.220446e
##           73           74           75           76           77           78           79 
## 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 1.000000e+00 1.000000e
##           81           82           83           84           85           86           87 
## 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e
##           89           90           91           92           93           94           95 
## 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e+00 1.000000e+00 2.220446e-16 2.220446e
##           97           98           99          100          101          102          103 
## 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e
##          105          106          107          108          109          110          111 
## 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e
##          113          114          115          116          117          118          119 
## 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e
##          121          122          123          124          125          126          127 
## 1.000000e+00 2.220446e-16 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e
##          129          130          131          132          133          134          135 
## 2.220446e-16 2.220446e-16 2.220446e-16 1.000000e+00 2.220446e-16 1.000000e+00 2.220446e-16 2.220446e
##          137 
## 1.000000e+00
```

```r
# Multinomial model:
# Example: Suppose we want to predict the Destination of a traveler's trip based on their age, gender, 
# Loading the nnet package
library(nnet)
Accommodation <- data$Accommodation.type
Gender <- data$Traveler.gender
Destination <- data$Destination
Age <- data$Traveler.age
dest_model <- multinom(Destination ~ Age + Gender + Accommodation, data = data)
```

```
## # weights:  295 (232 variable)
## initial  value 558.622630
## iter  10 value 522.377509
## iter  20 value 505.690954
## iter  30 value 490.125369
## iter  40 value 477.196263
## iter  50 value 461.183787
## iter  60 value 445.047376
## iter  70 value 433.448886
## iter  80 value 425.758675
## iter  90 value 423.896479
## iter 100 value 422.905806
## final  value 422.905806
## stopped after 100 iterations
```

```r
dest_model_purpose <- predict(dest_model, type = "class")
print(dest_model_purpose)
```

```
##   [1] Paris                  Bali                   Bangkok                Paris
##   [5] Tokyo, Japan           Paris, France          Bali                   Tokyo, Japan
##   [9] Paris                  Bali                   Tokyo, Japan           Bali
##  [13] Bali                   London, UK             Tokyo, Japan           Paris
##  [17] Paris                  Bali                   Paris                  Paris, France
##  [21] Tokyo, Japan           Paris                  Bali                   Bali
##  [25] Dubai                  Bali                   Paris                  Paris
##  [29] Bangkok                Tokyo, Japan           Tokyo, Japan           Bali
##  [33] Greece                 Paris, France          Bangkok                Bali
##  [37] Tokyo, Japan           Bali                   Dubai                  Bali
##  [41] Bali                   Bangkok                Paris                  Cape Town, South Afric
##  [45] Tokyo, Japan           Paris, France          Bali                   Tokyo, Japan
##  [49] Paris                  Paris                  Tokyo, Japan           Tokyo, Japan
##  [53] Tokyo, Japan           Bali                   France                 Paris
##  [57] Bali                   Tokyo, Japan           Dubai                  Bali
##  [61] Paris                  Bali                   Bali                   Paris, France
##  [65] Bali                   Tokyo, Japan           Paris, France          Tokyo, Japan
##  [69] Bali                   Paris, France          Bali                   Bali
##  [73] Paris                  Bangkok                Paris, France          Bali
##  [77] Tokyo, Japan           Paris, France          Tokyo, Japan           Berlin, Germany
##  [81] Bali                   Paris, France          Paris                  Bali
##  [85] Bangkok                Paris, France          Bali                   Bali
##  [89] Dubai                  Tokyo, Japan           Bali                   Bali
##  [93] Tokyo, Japan           Paris                  Bali                   Bangkok
```

```
##  [97] Paris, France          Cape Town, South Africa Paris                   Bangkok
## [101] Tokyo, Japan           Paris                   Bali                    Paris, France
## [105] Paris                  Bali                    Paris, France           Bali
## [109] Tokyo, Japan           Paris, France           Bangkok                 Paris
## [113] Tokyo, Japan           Paris                   Bali                    Paris
## [117] Paris, France          Tokyo, Japan            Paris, France           Bali
## [121] Paris, France          Bali                    Dubai                   Bali
## [125] Bali                   Paris                   Paris                   Bali
## [129] Bangkok                Paris                   Bangkok                 Tokyo, Japan
## [133] Bali                   Paris                   Bali                    Tokyo, Japan
## [137] Paris, France
## 59 Levels: Amsterdam Amsterdam, Netherlands Athens, Greece Auckland, New Zealand Australia ... Vanco
```

```
# Poisson model:
# Example: Suppose we want to predict the cost of visit to a certain destination in a year based on the
Cost <- data$Transportation.cost
cost_model <- glm(Cost ~ Age + Destination, data = data, family = poisson())
cost_prediction <- predict(visit_model, type = "response")
print(cost_prediction)
```

```
##         1         2         3         4         5         6         7         8         9        10
##  298.9452  500.0000  764.6508  928.4040  613.8233  874.0461 1450.5088  860.0040  450.0000  800.0000
##        11        12        13        14        15        16        17        18        19        20
##  596.2958  865.5685 1200.0000  700.0000  400.0000  150.0000  513.3954  840.9545  350.0455  423.0153
##        21        22        23        24        25        26        27        28        29        30
##  318.6510  897.0595  335.3982  350.8744  511.7762  800.0000  425.0000  500.0000  150.0000  300.0000
##        31        32        33        34        35        36        37        38        39        40
##  500.0000   50.0000  600.0000  100.0000  800.0000  120.0000  400.0000   75.0000  866.2582 1450.5088
##        41        42        43        44        45        46        47        48        49        50
##  614.4700  601.9735  862.2708  302.1096  861.4769 1505.5280  366.7300 1572.9270  450.0000  868.9989
##        51        52        53        54        55        56        57        58        59        60
##  613.5003  200.0000  927.9154  862.2708  365.5734  298.9452  862.3844   20.0000  300.0000  697.9748
##        61        62        63        64        65        66        67        68        69        70
##  514.7486  419.2462  850.4475  902.2697  425.0000  840.5119  515.8338  318.3157  349.5837  337.5239
##        71        72        73        74        75        76        77        78        79        80
##  844.9487  760.6357  614.7936  601.0235  875.8887  300.0000  756.6416  869.7109  400.0000  943.6806
##        81        82        83        84        85        86        87        88        89        90
##  800.0000  700.9207  514.7486  419.2462  847.1758  903.2202  319.3228  348.9415  549.2761  348.2978
##        91        92        93        94        95        96        97        98        99       100
##  335.5748  843.1712  150.0000  514.7486  419.2462  903.2202  350.4143  322.5330  335.7516  351.2441
##       101       102       103       104       105       106       107       108       109       110
##  849.5525  400.0000  841.3974  150.0000  514.7486  419.2462  321.1774  841.8405  894.2303  350.5988
##       111       112       113       114       115       116       117       118       119       120
##  700.0000  335.7516  150.0000  550.7239  500.0000  615.7654  876.3499  864.2336  699.8146  758.2367
##       121       122       123       124       125       126       127       128       129       130
##  600.7072  900.0000  613.1774  866.0555  867.6275  759.8352  870.8308  614.4700 2000.0000 1448.9823
##       131       132       133       134       135       136       137
##  701.2899 1494.4720  865.4544 1577.0730  367.6966  860.5702 2500.0000
```

```
###############################################################################################
## Question 3
###############################################################################################
# (a) Consider two categorical variables of the dataset, develop a binary decision
```

```r
# making strategy to check whether two variables are independent at the significant level alpha=0.01

# Dataset Source: https://www.kaggle.com/datasets/rkiattisak/traveler-trip-data
library(help = "graphics")
data <- read.csv("Travel details dataset.csv")

##  i). Stating the hypotheses
# Null hypothesis : Two categorical variables selected, Traveler gender and Accommodation type are inde
# at the significant level alpha=0.01
# Alternative hypothesis :Two categorical variables selected, Traveler gender and Accommodation type ar

# ii). Find statistic and critical values

# Chi-square test is the best method to define the statistical relationship between two categorical var
# First calculating chi-square statistic and compare it to the critical value from the chi-square distr

# Contingency table of Traveler gender and Accommodation type from data frame data
vGender <- data$Traveler.gender
vAccType <- data$Accommodation.type
c_table <- table(vGender,vAccType)
cat("Contingency Table:", c_table, "\n")
```

```
## Contingency Table: 18 12 0 1 9 15 30 30 9 5 1 0 2 1 1 3
```

```r
# chi-square test against c_table
chsq_test <- chisq.test(c_table)
```

```
## Warning in chisq.test(c_table): Chi-squared approximation may be incorrect
```

```r
# Finding chi-square statistic and p-value.P-value is the area under the density curve of chi-square di
# to the right of the value of the test statistic.P means the probability, for a given statistical mode

chsq_statistic <- chsq_test$statistic
p_value <- chsq_test$p.value

cat("chi-square statistic:", chsq_statistic, "\n")
```

```
## chi-square statistic: 7.113908
```

```r
cat("p_value:", p_value, "\n")
```

```
## p_value: 0.417117
```

```r
# Finding critical value. In hypothesis tests, Critical value implies boundary of how extreme
# a test statistic we need to reject the Null hypothesis.
# To determine this we need to have Number of degrees of freedom,Number and type of Tails,The level of

rows <- nrow(c_table)
cols <- ncol(c_table)
df <- (rows - 1) * (cols - 1)
critical_value <- qchisq(0.99, df)
cat("critical_value:", critical_value, "\n")
```

```
## critical_value: 18.47531
```

```r
# iii). Explain your decision and Interpret results

# Here the chi-square Statistic is 7.113908 , Probability or P_Value is 0.417117  and Critical value is
# Decision making is based on that if chi-square statistic is greater than critical value then we have
# Here we do not require to reject the Null hypothesis since the values obtained are justifying our Nul
# So the conclusion is the Selected two categorical variables are independent at the 0.01 level of sign
# ie  p-value(0.417117) is greater than the level of significance (0.01), and it supports our null hypo

if (chsq_statistic > critical_value) {
  cat(" The two categorical variables,Traveler gender and Accommodation type are not independent thus r
} else {
  cat("The two categorical variables,Traveler gender and Accommodation type are independent at the 0.01
}
```

```
## The two categorical variables,Traveler gender and Accommodation type are independent at the 0.01 leve
```

```r
##################################################################################################
# (b) Consider one categorical variable, apply goodness of fit test to evaluate whether a
# candidate set of probabilities can be appropriate to quantify the uncertainty of class frequency
# at the significant level alpha=0.05

# Null hypothesis: The observed class frequencies fit the candidate set of probabilities.
# Alternative hypothesis: The observed class frequencies do not fit the candidate set of probabilities.
# Calculating chi-square statistic and comparing it with critical value from the chi-square distributio
# where k is the number of categories in the categorical variable, at the alpha level of significance 0
# If the chi-square statistic is greater than the critical value, we reject the null hypothesis.

# Calculate the observed frequencies
obs_freq <- table(data$Transportation.type)

# Set the candidate set of probabilities
p <- c(0.5, 0.3, 0.1, 0.1)

# Calculate the expected frequencies
exp_freq <- sum(obs_freq) * p

# Calculate the chi-squared test statistic
chi_sq <- sum((obs_freq - exp_freq)^2 / exp_freq)
```

```
## Warning in obs_freq - exp_freq: longer object length is not a multiple of shorter object length

## Warning in (obs_freq - exp_freq)^2/exp_freq: longer object length is not a multiple of shorter objec
## length
```

```r
# Calculate the degrees of freedom
df <- length(p) - 1

# Calculate the p-value
p_value <- 1 - pchisq(chi_sq, df)
```

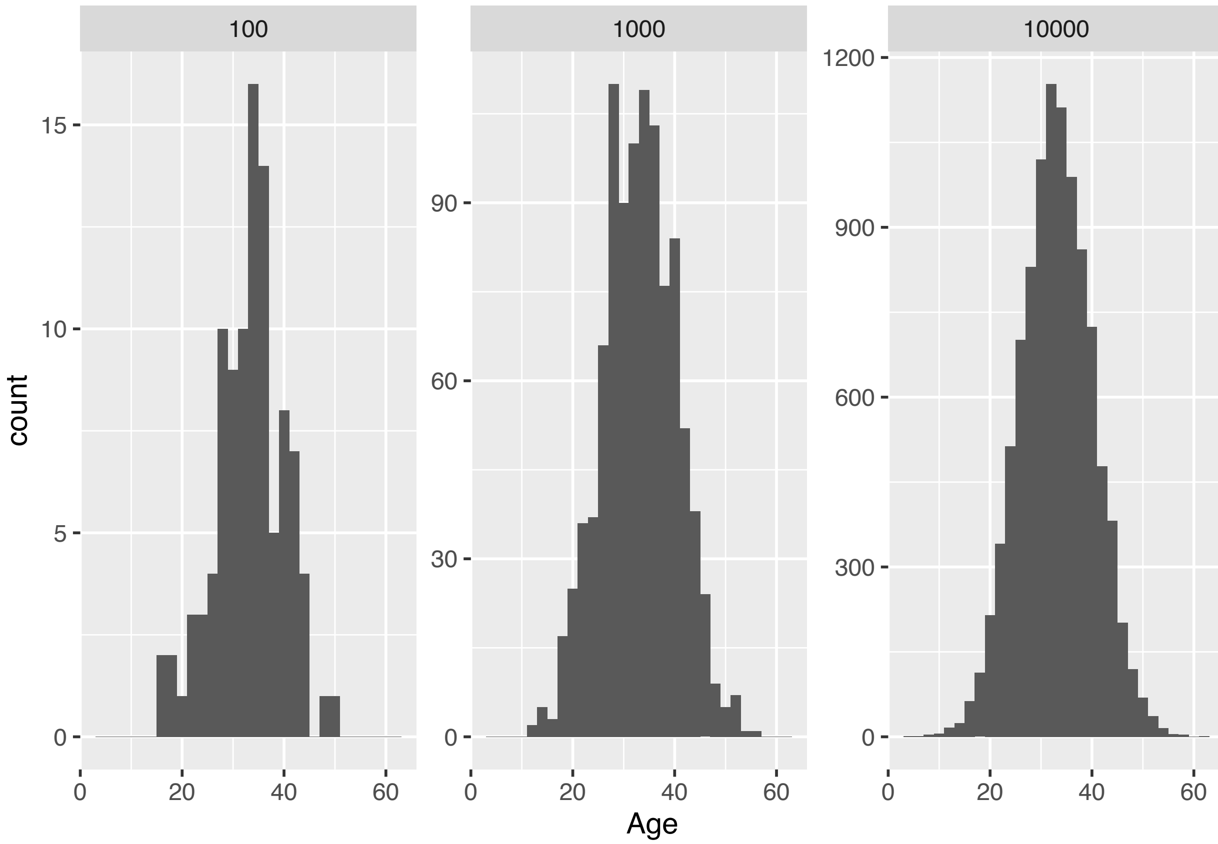
```r
# Compare the p-value to the significance level alpha
alpha <- 0.05
if (p_value < alpha) {
  print("Reject null hypothesis")
} else {
  print("Fail to reject null hypothesis")
}
```

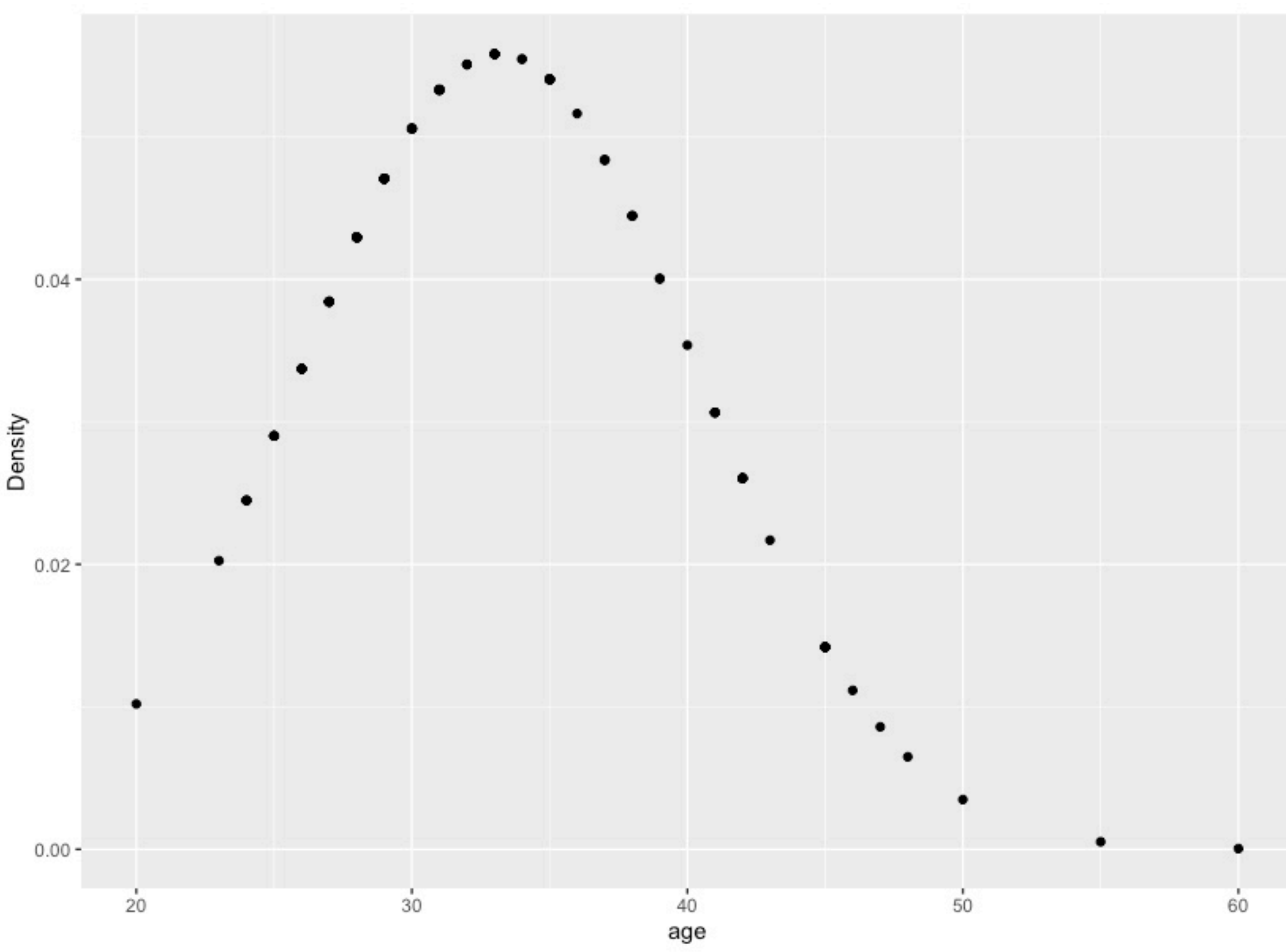
```
## [1] "Reject null hypothesis"
```

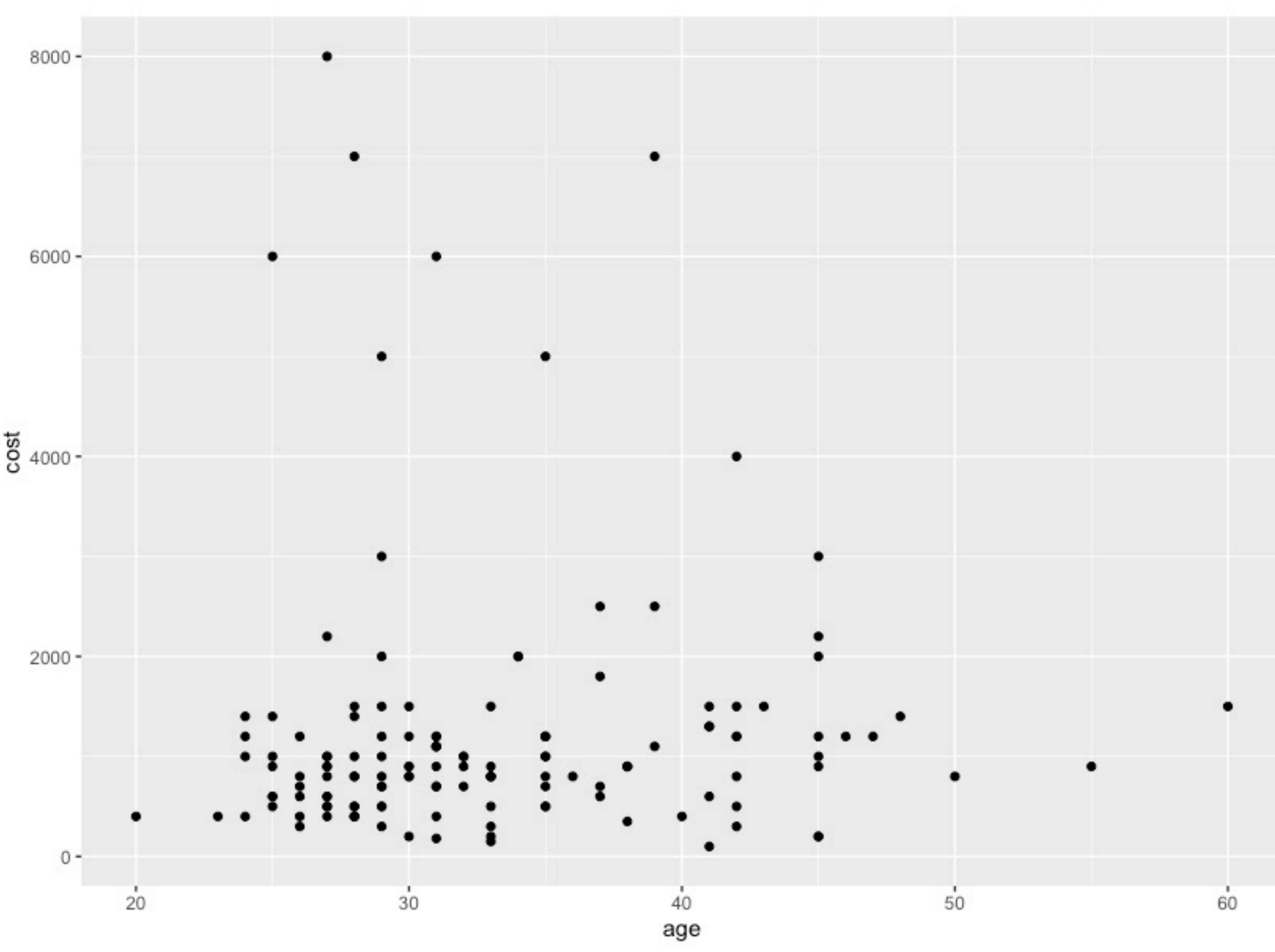
```r
###################################################################################################
# (c) Consider one continuous variable in the dataset, and apply test of mean for a proposed candidate

age <- data$Traveler.age
# Set the proposed candidate mean
mue_value <- 40

# Calculate the sample mean and sample standard deviation
age_mean <- mean(age)
age_sd <- sd(age)

# Calculate the t-test statistic
t_stat <- (age_mean - mue_value) / (age_sd / sqrt(length(age)))

# Calculate the degrees of freedom
freedom <- length(age) - 1

# Calculate the p-value
p_value <- 2 * pt(abs(t_stat), freedom, lower.tail = FALSE)

# Comparing the p-value to the significance level alpha as per question
alpha <- 0.05
if (p_value < alpha) {
  print("Comment: Rejecting null hypothesis")
} else {
  print("Comment: Can not reject null hypothesis")
}
```

```
## [1] "Comment: Rejecting null hypothesis"
```

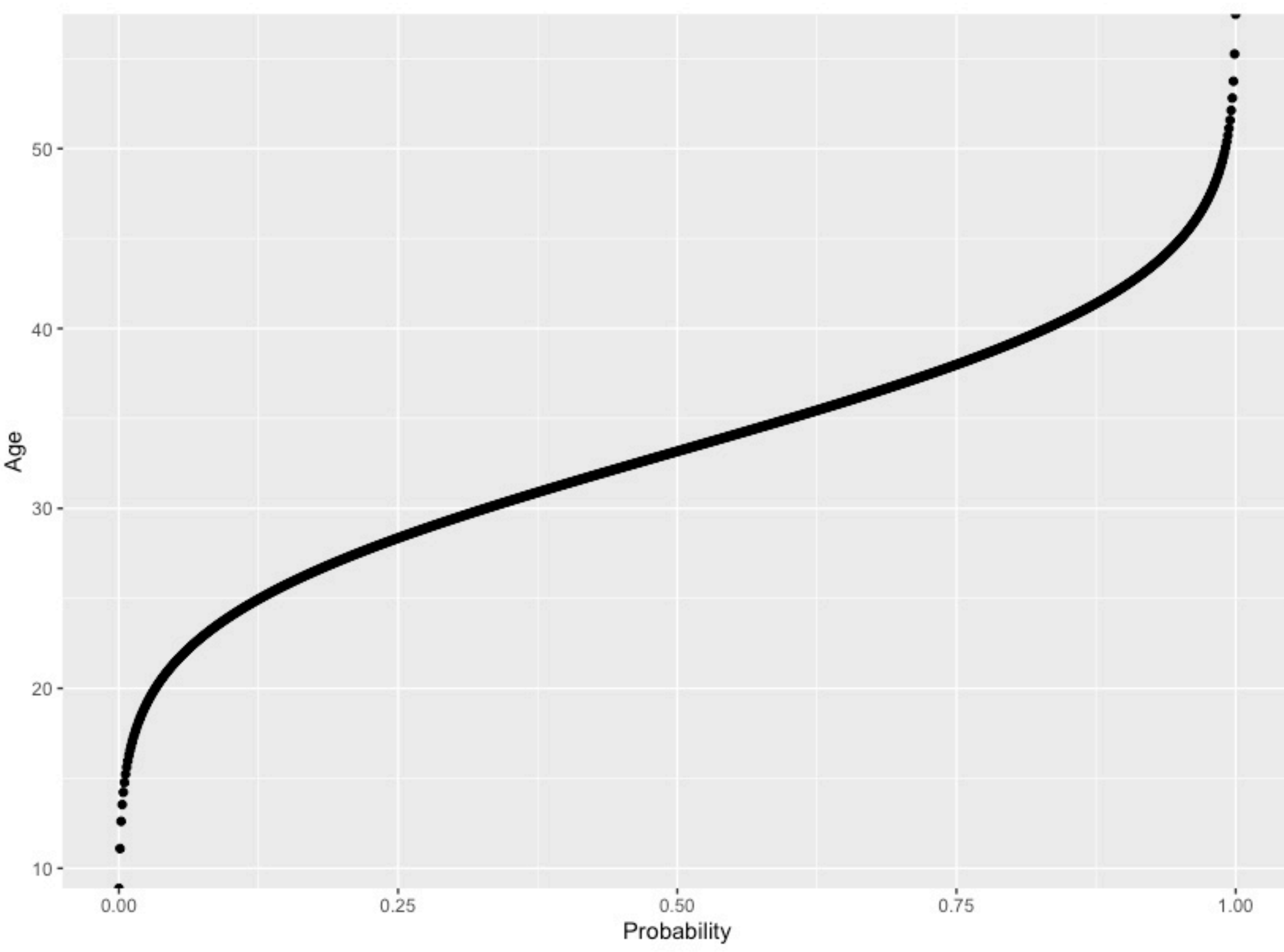
```r
###################################################################################################
```


100
1000
10000
count
Age
0
5
10
15
30
60
90
300
600
900
1200
0
20
40
60


Density
0.00
0.02
0.04
20
30
40
50
60
age



8000
6000
4000
2000
0
cost
20
30
40
50
60
age


Age
50
40
30
20
10
0.00
0.25
0.50
0.75
1.00
Probability