Exploring PUMS for NJ

Use shapiro test to assess normality

```
shapiro = function(x) {
    return(shapiro.test(x)$p.value)
}

ks = function(x) {
    return(ks.test(x, "pnorm", mean = mean(x), sd = sd(x))$p.value)
}
```

Read in NJ Pums data

```
pumsnj = read.csv("psam_h34.csv")

# Some numbers for house values
sum(!is.na(pumsnj$VALP))  #24423 non NA house values in NJ
```

```
## [1] 24423
```

```
mean(sapply(split(pumsnj, pumsnj$PUMA), function(x) {
    sum(is.na(x$VALP))
}))  #average 211 people per PUMA in NJ
```

```
## [1] 211.4521
```

```
length((sapply(split(pumsnj, pumsnj$PUMA), function(x) {
    sum(is.na(x$VALP))
})))  #73 different PUMAS in NJ, but 21 counties
```

```
## [1] 73
```

```
# Some numbers for household income
sum(!is.na(pumsnj$HINCP))  #33660 non NA house values in NJ
```

```
## [1] 33660
```

```
mean(sapply(split(pumsnj, pumsnj$PUMA), function(x) {
    sum(is.na(x$HINCP))
}))  #average 85 people per PUMA in NJ
```

```
## [1] 84.91781
```

```
length((sapply(split(pumsnj, pumsnj$PUMA), function(x) {
    sum(is.na(x$HINCP))
})))  #73 different PUMAS in NJ, but 21 counties
```

```
## [1] 73
```

Normality within PUMAs? NO

```
# Examine normality of house values within PUMAs
pvals = sapply(split(log(pumsnj$VALP), pumsnj$PUMA), shapiro)
mean(pvals > 0.05, na.rm = T)  #all rejected
```

```
## [1] 0
```

```
# Examine normality of household incomes within PUMAs
pvals = sapply(split(log(pumsnj$HINCP + 1), pumsnj$PUMA), shapiro)  # some are 0 so offset by 1 then lo
```

```
## Warning in log(pumsnj$HINCP + 1): NaNs produced
```

```
mean(pvals > 0.05, na.rm = T)  #all rejected
```

```
## [1] 0
```

Link PUMAs to FIPS in NJ (for now just assume PUMAs are smaller than FIPS)

Using equivalency at https://www2.census.gov/geo/pdfs/reference/puma/2010_PUMA_Equivalency_Summary_Levels.pdf

```
library(gdata)
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.

##

## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.

##
## Attaching package: 'gdata'

## The following object is masked from 'package:stats':
##
##     nobs

## The following object is masked from 'package:utils':
##
##     object.size

## The following object is masked from 'package:base':
##
##     startsWith
```

```
equiv <- read.delim("PUMSEQ10_34.txt", stringsAsFactor = FALSE,
    header = F)
head(equiv)
```

```
##
## 1 795340177979500101                                                    112887    62119Atlantic County (Eas
## 2 796340177979500101001008822 70                                        112887    62119Atlantic County (pa
## 3 797340177979500101001008822700000000000000                                0        0County subdivisions
## 4 798340177979500101001008822700000000000000999999999999                    0        0County subdivisions
## 5 799340177979500101001008822700000000000000999999999999010101              0        0Census Tract 101.01
## 6 799340177979500101001008822700000000000000999999999999010102              0        0Census Tract 101.02
```

```
# Messy.  example: 796 34 0177797950 0101 001 00882270 796
# state random puma

codes = unlist(lapply(strsplit(equiv[, 1][which(startsWith(equiv[,
    1], "796") == T)], "\\s+"), function(x) {
    x[1]
}))
# Extract PUMAs
pumas = as.numeric(substring(codes, 15, 18))
```

```
# Extract FIPS
fips = paste(substring(codes, 4, 5), substring(codes, 19, 21),
    sep = "")
# Dataframe matchig PUMAs to FIPS
puma.fips.match = data.frame(pumas, fips)

# Add fips to previous dataframe
pumsnj$fips = puma.fips.match$fips[match(pumsnj$PUMA, puma.fips.match$pumas)]
length(unique(pumsnj$fips))  #2 counties of 21 excluded from pums sample?? Is this because multiple sma
```

```
## [1] 19
```

```
# Number of individuals sampled from each fips
sapply(split(log(pumsnj$VALP), pumsnj$fips), length)
```

```
## 34001 34003 34005 34007 34011 34013 34015 34017 34019 34021 34023 34025 34027
##  2585  3700  2324  2281  1210  3230  1360  2678   740  1538  3542  2917  2168
## 34029 34031 34035 34037 34039 34041
##  3098  1710  1356   811  2020   591
```

Normality within FIPS?? NO

```
# Examine normality of house values within fips
pvals = sapply(split(log(pumsnj$VALP), pumsnj$fips), shapiro)
mean(pvals > 0.05, na.rm = T)
```

```
## [1] 0
```

```
# Examine normality of household incomes within fips
pvals = sapply(split(log(pumsnj$HINCP + 1), pumsnj$fips), shapiro)  # some are 0 so offset by 1 then lo
```

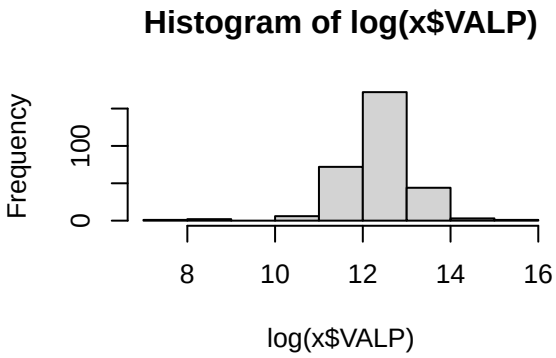
```
## Warning in log(pumsnj$HINCP + 1): NaNs produced
```

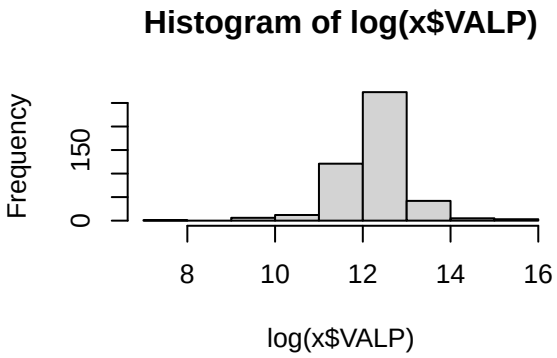
```
mean(pvals > 0.05, na.rm = T)
```

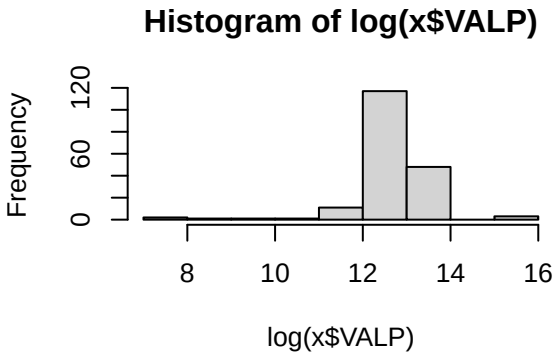
```
## [1] 0
```

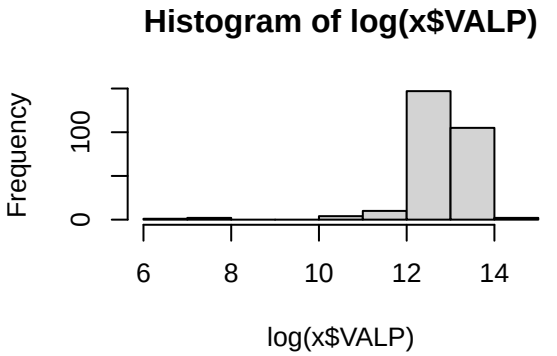
Visually assess normality

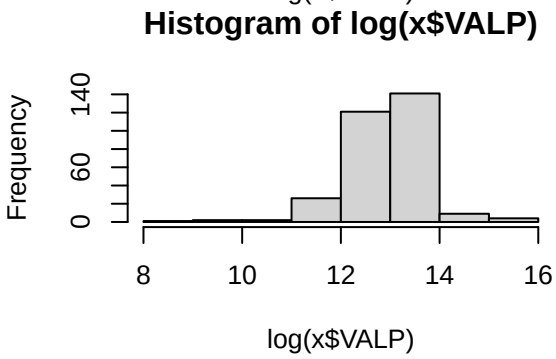
```
# Within PUMA
par(mfrow = c(2, 2))
sapply(split(pumsnj, pumsnj$PUMA), function(x) {
    hist(log(x$VALP))
})
```

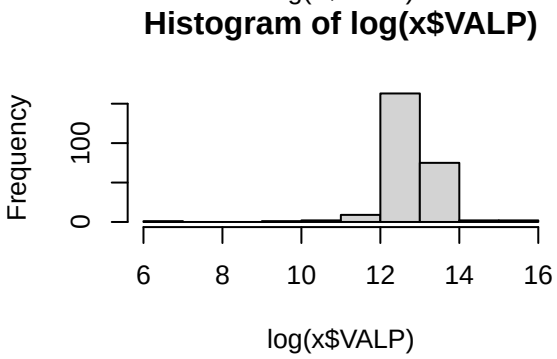

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

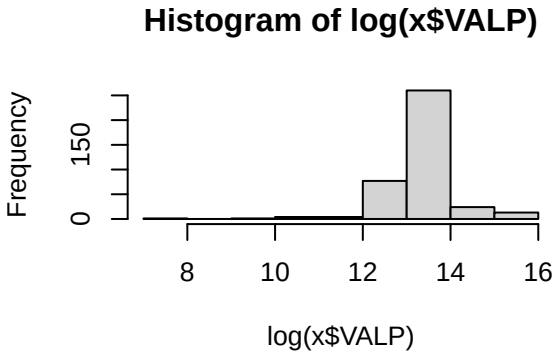

Histogram of log(x$VALP)
Frequency
0
150
8
10
12
14
16
log(x$VALP)

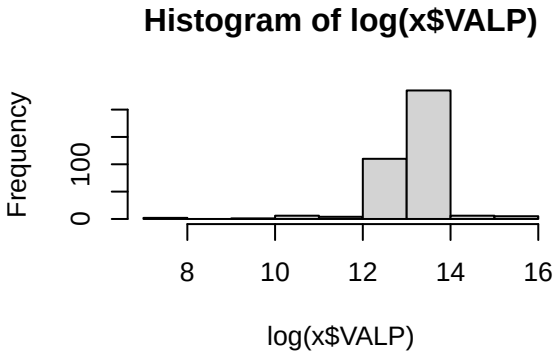

Histogram of log(x$VALP)
Frequency
0
60
120
8
10
12
14
16
log(x$VALP)

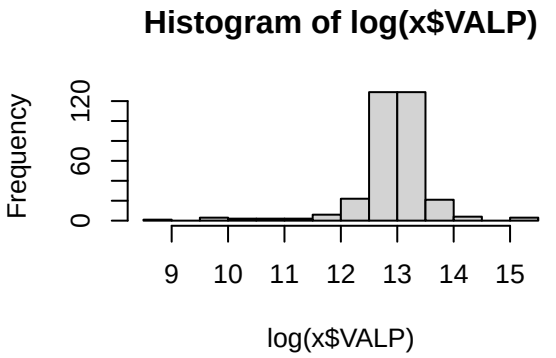

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
log(x$VALP)

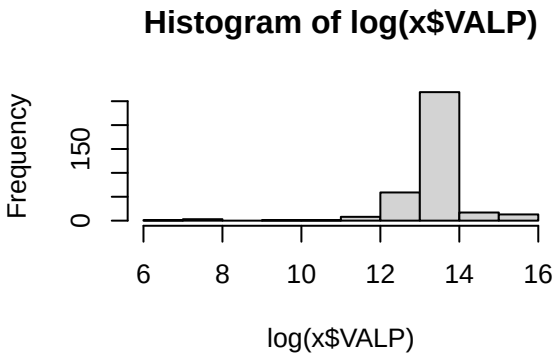

Histogram of log(x$VALP)
Frequency
0
60
140
8
10
12
14
16
log(x$VALP)

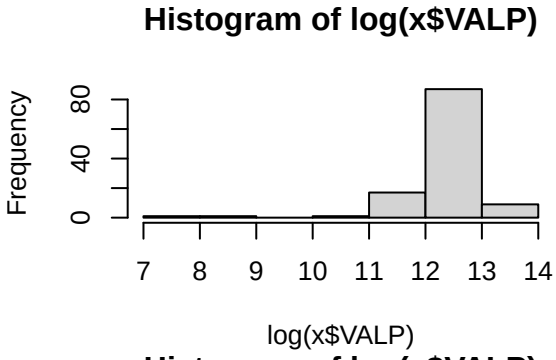

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
16
log(x$VALP)

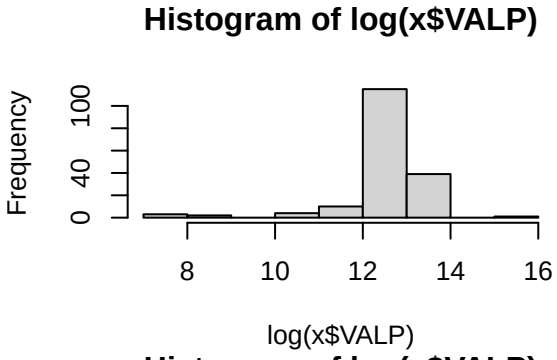

Histogram of log(x$VALP)
Frequency
0
150
8
10
12
14
16
log(x$VALP)

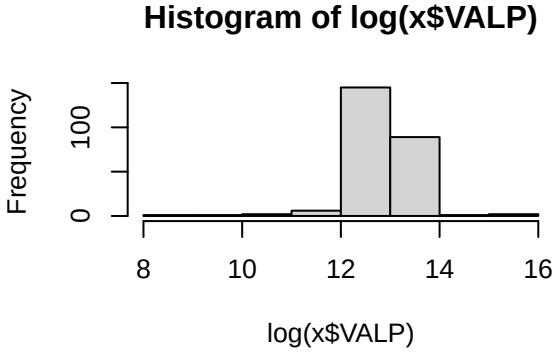

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

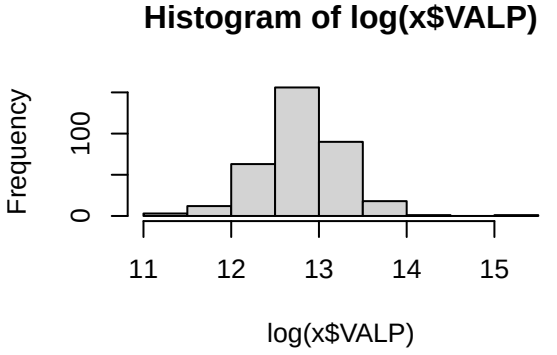

Histogram of log(x$VALP)
Frequency
0
60
120
9
10
11
12
13
14
15
log(x$VALP)

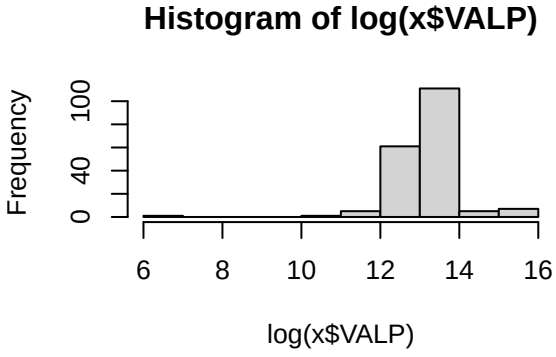

Histogram of log(x$VALP)
Frequency
0
150
6
8
10
12
14
16
log(x$VALP)

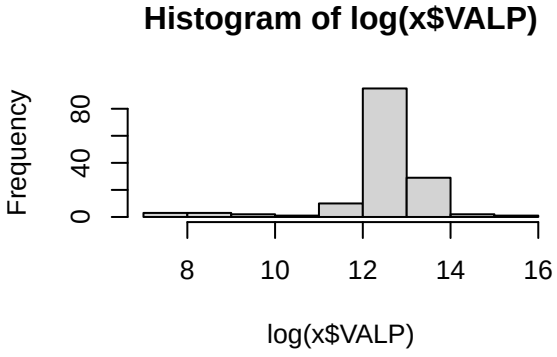

Histogram of log(x$VALP)
Frequency
0
40
80
7
8
9
10
11
12
13
14
log(x$VALP)

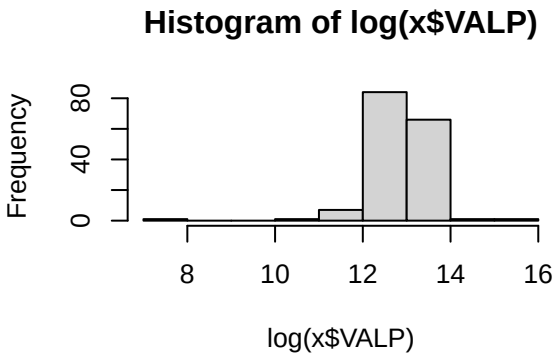

Histogram of log(x$VALP)
Frequency
0
40
100
8
10
12
14
16
log(x$VALP)

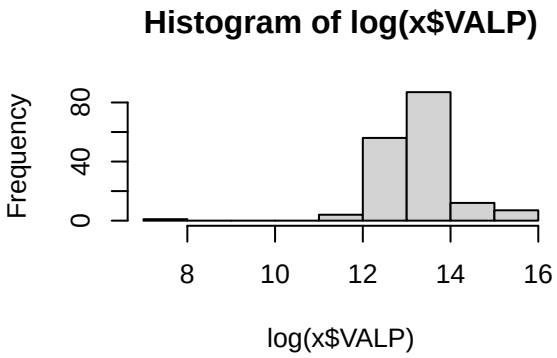

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

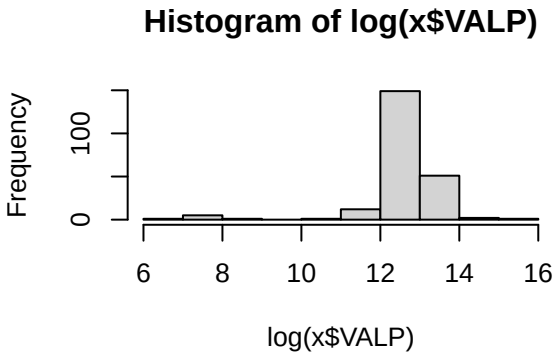

Histogram of log(x$VALP)
Frequency
0
100
11
12
13
14
15
log(x$VALP)

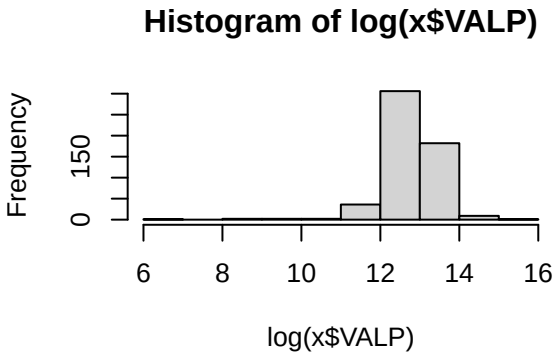

Histogram of log(x$VALP)
Frequency
0
40
100
6
8
10
12
14
16
log(x$VALP)

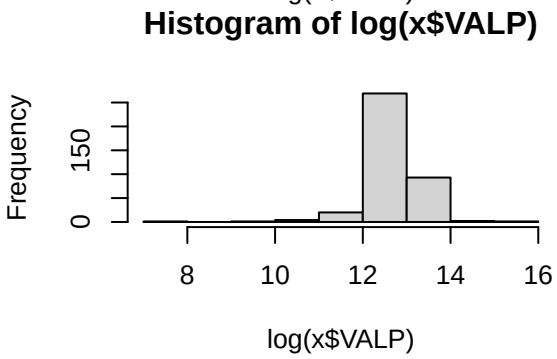

Histogram of log(x$VALP)
Frequency
0
40
80
8
10
12
14
16
log(x$VALP)

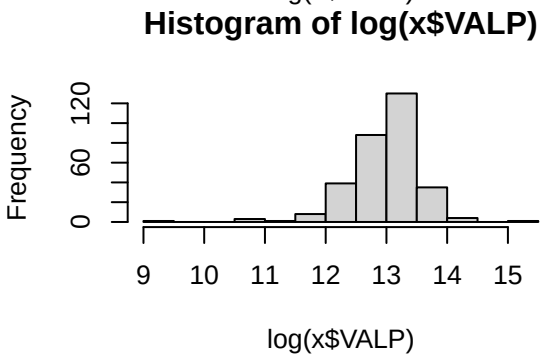

Histogram of log(x$VALP)
Frequency
0
40
80
8
10
12
14
16
log(x$VALP)

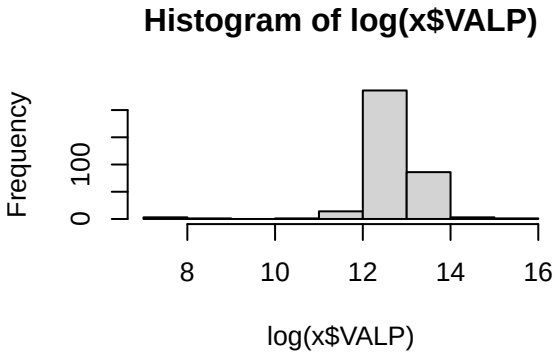

Histogram of log(x$VALP)
Frequency
0
40
80
8
10
12
14
16
log(x$VALP)

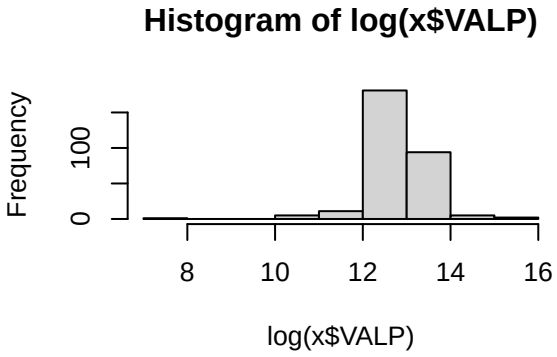

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
16
log(x$VALP)

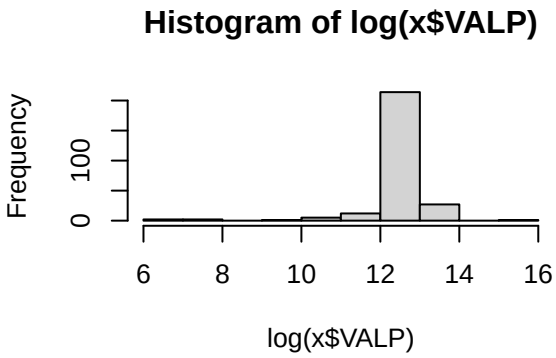

Histogram of log(x$VALP)
Frequency
0
150
6
8
10
12
14
16
log(x$VALP)

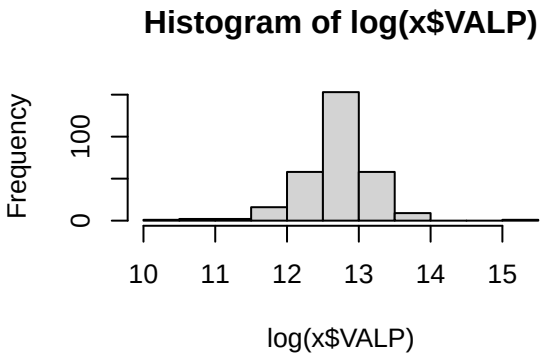

Histogram of log(x$VALP)
Frequency
0
150
8
10
12
14
16
log(x$VALP)

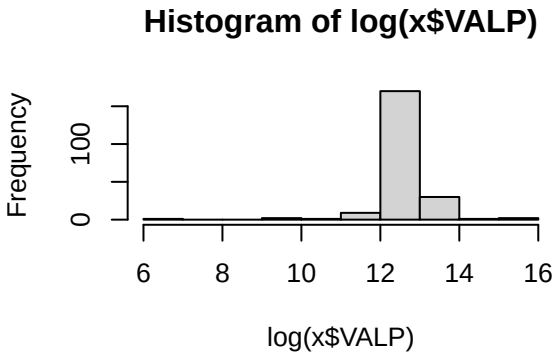

Histogram of log(x$VALP)
Frequency
0
60
120
9
10
11
12
13
14
15
log(x$VALP)

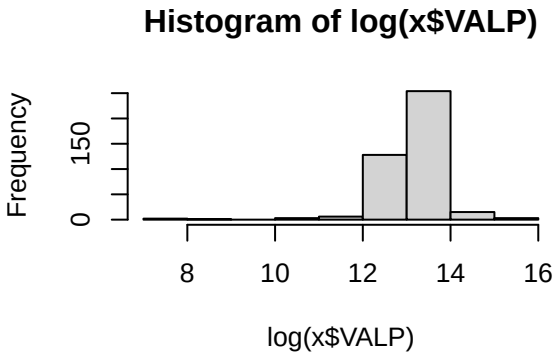

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

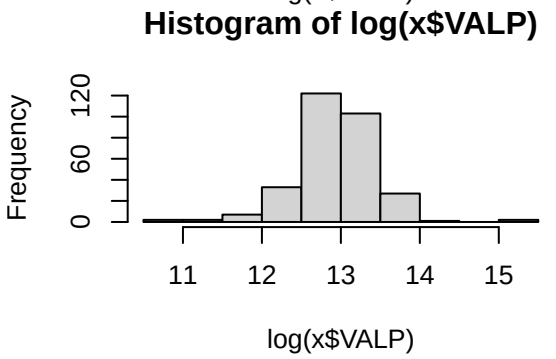

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

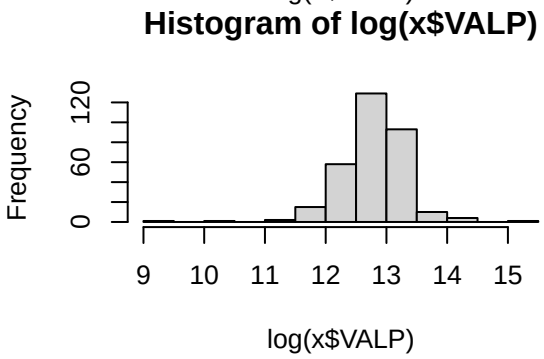

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
16
log(x$VALP)

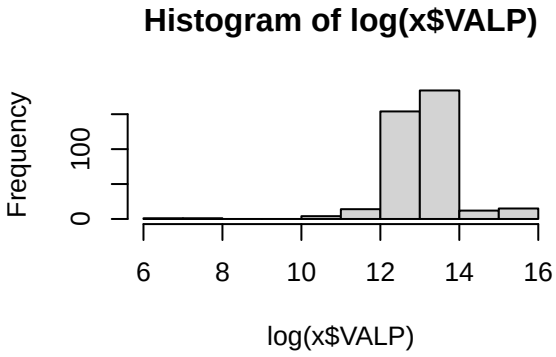

Histogram of log(x$VALP)
Frequency
0
100
10
11
12
13
14
15
log(x$VALP)

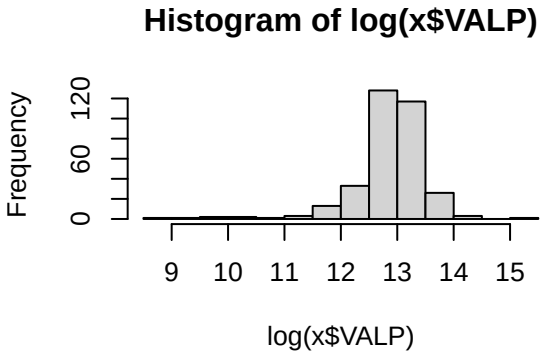

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
16
log(x$VALP)

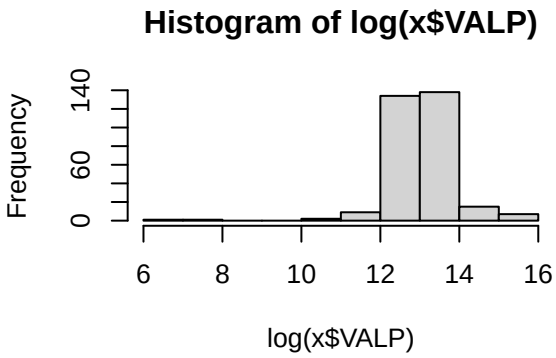

Histogram of log(x$VALP)
Frequency
0
150
8
10
12
14
16
log(x$VALP)

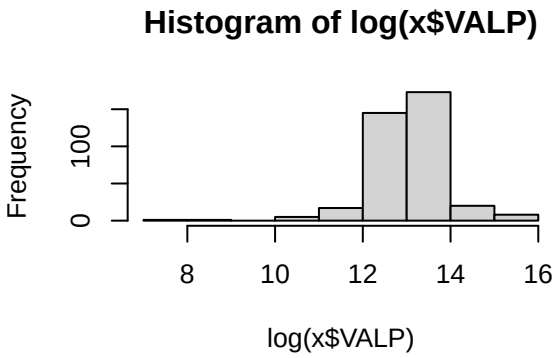

Histogram of log(x$VALP)
Frequency
0
60
120
11
12
13
14
15
log(x$VALP)

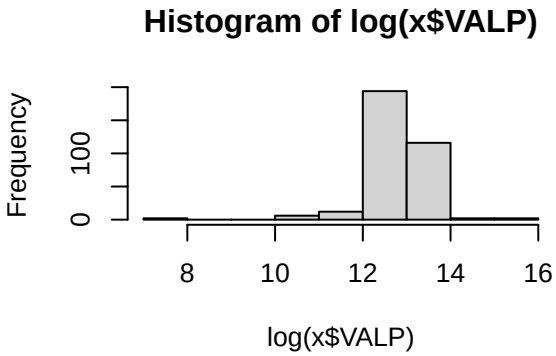

Histogram of log(x$VALP)
Frequency
0
60
120
9
10
11
12
13
14
15
log(x$VALP)

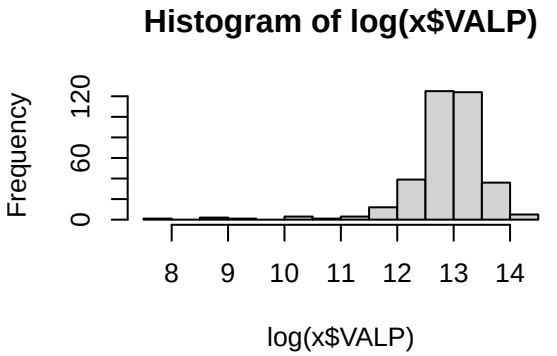

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
16
log(x$VALP)

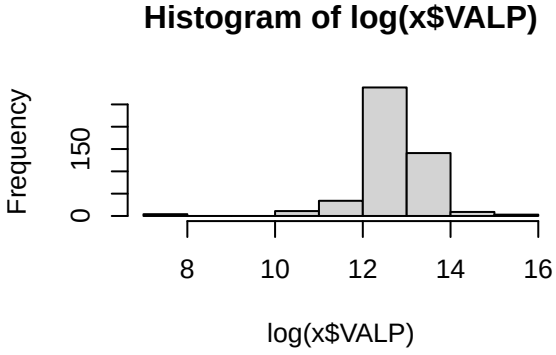

Histogram of log(x$VALP)
Frequency
0
60
120
9
10
11
12
13
14
15
log(x$VALP)

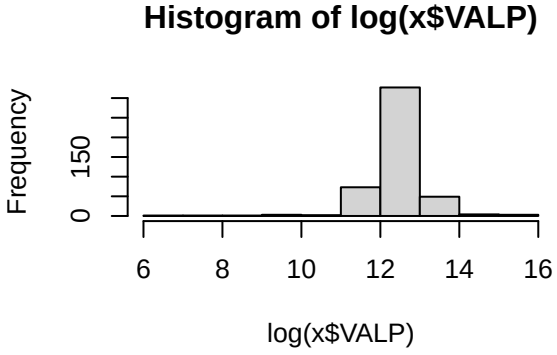

Histogram of log(x$VALP)
Frequency
0
60
140
6
8
10
12
14
16
log(x$VALP)

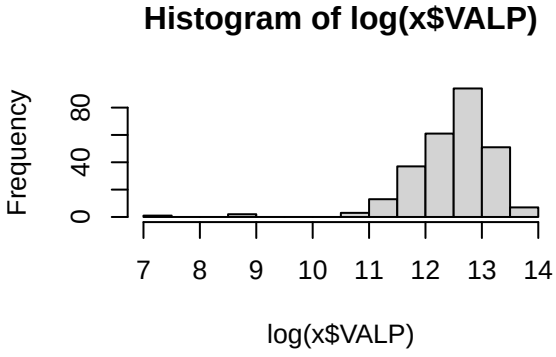

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

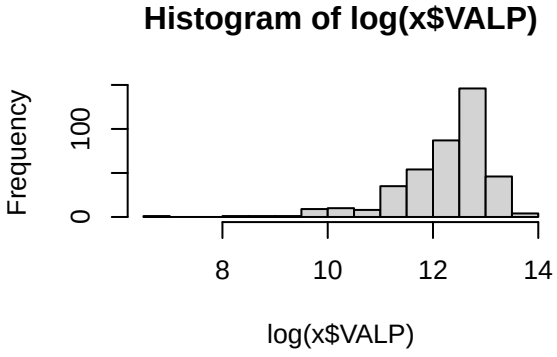

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

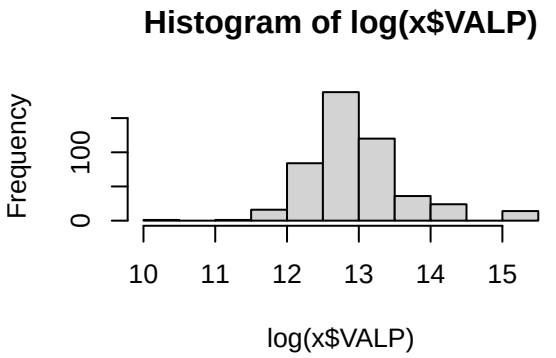

Histogram of log(x$VALP)
Frequency
0
60
120
8
9
10
11
12
13
14
log(x$VALP)

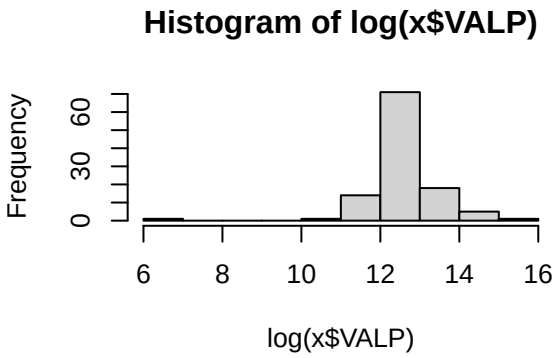

Histogram of log(x$VALP)
Frequency
0
150
8
10
12
14
16
log(x$VALP)

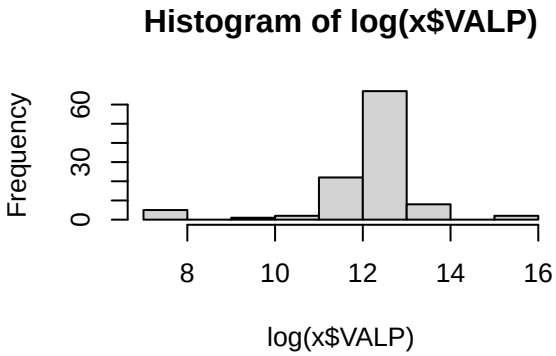

Histogram of log(x$VALP)
Frequency
0
150
6
8
10
12
14
16
log(x$VALP)

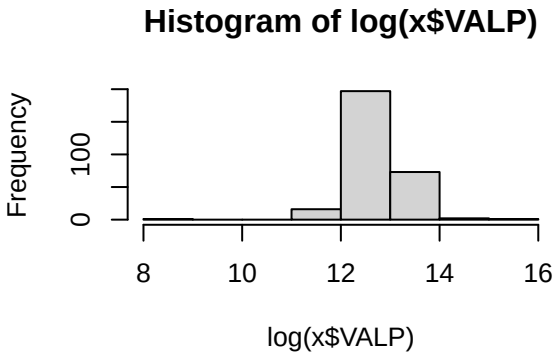

Histogram of log(x$VALP)
Frequency
0
40
80
7
8
9
10
11
12
13
14
log(x$VALP)

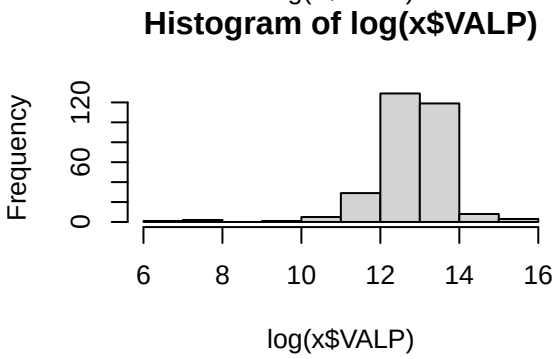

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
log(x$VALP)

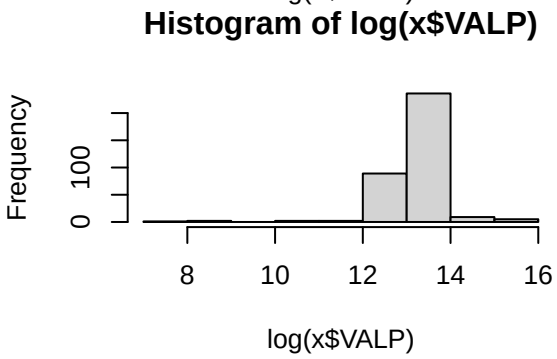

Histogram of log(x$VALP)
Frequency
0
100
10
11
12
13
14
15
log(x$VALP)

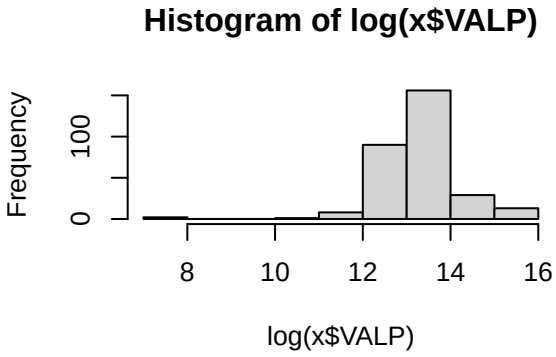

Histogram of log(x$VALP)
Frequency
0
30
60
6
8
10
12
14
16
log(x$VALP)

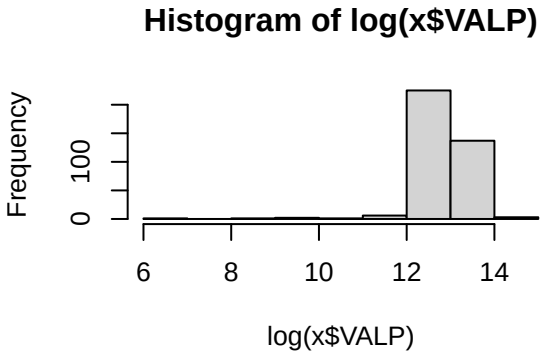

Histogram of log(x$VALP)
Frequency
0
30
60
8
10
12
14
16
log(x$VALP)

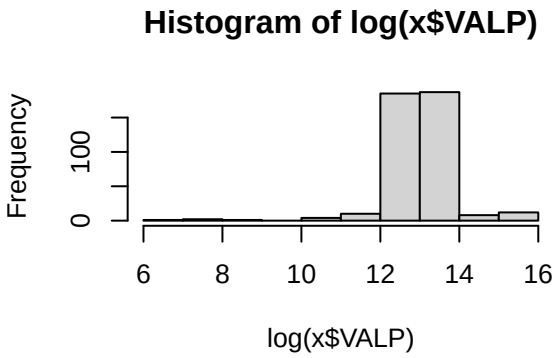

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

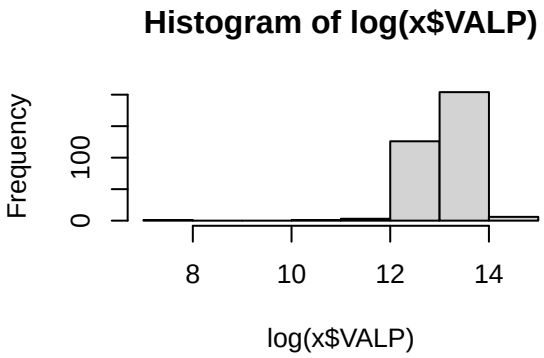

Histogram of log(x$VALP)
Frequency
0
60
120
6
8
10
12
14
16
log(x$VALP)

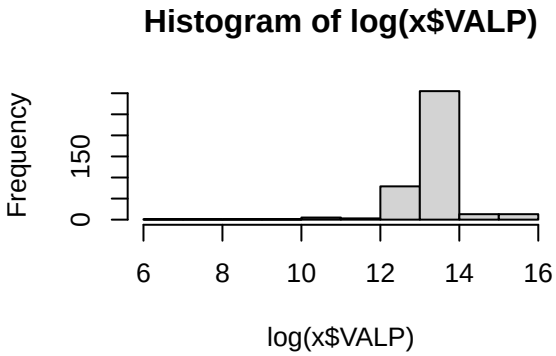

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

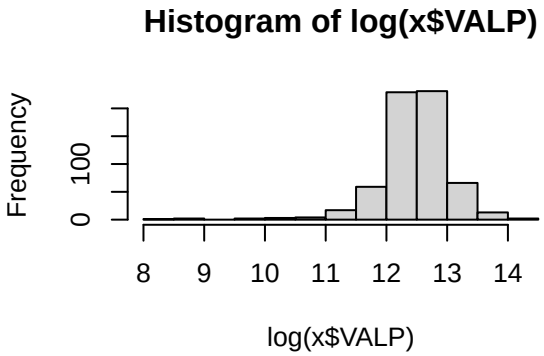

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

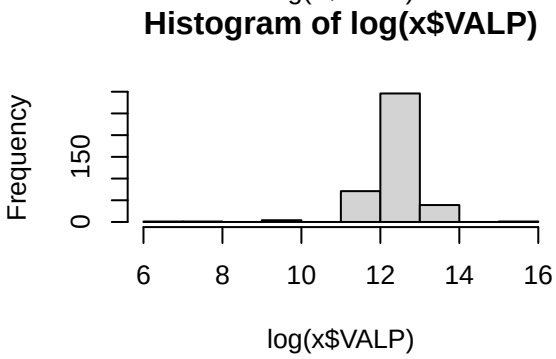

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
log(x$VALP)

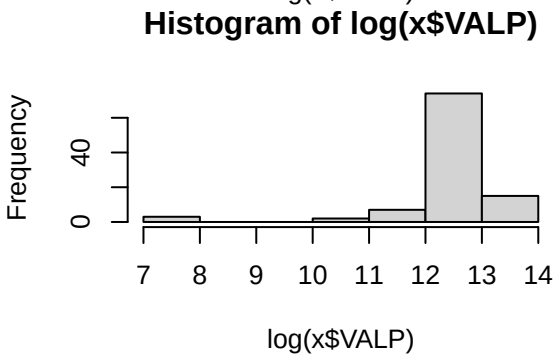

Histogram of log(x$VALP)
Frequency
0
100
6
8
10
12
14
16
log(x$VALP)

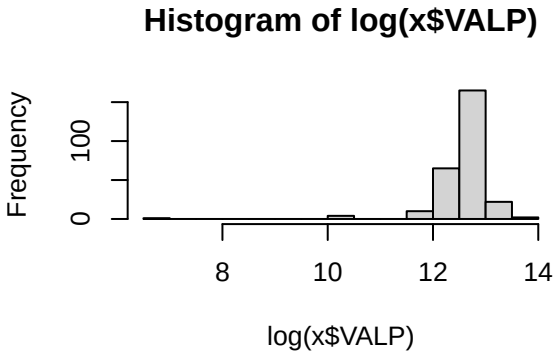

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
log(x$VALP)

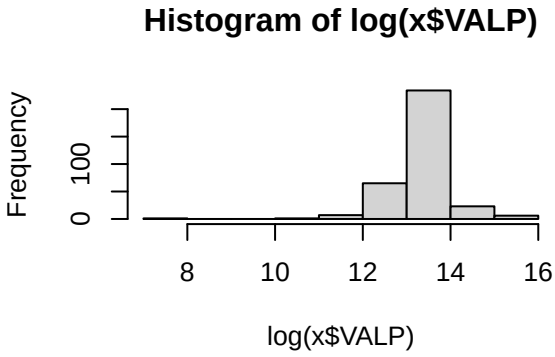

Histogram of log(x$VALP)
Frequency
0
150
6
8
10
12
14
16
log(x$VALP)

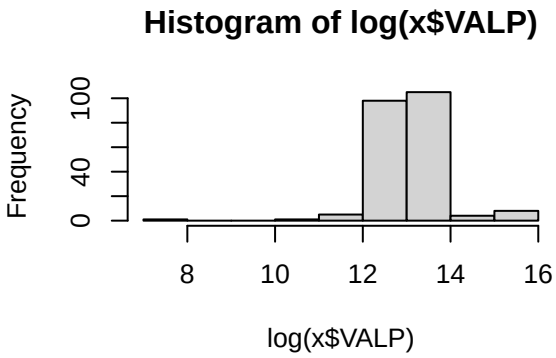

Histogram of log(x$VALP)
Frequency
0
100
8
9
10
11
12
13
14
log(x$VALP)

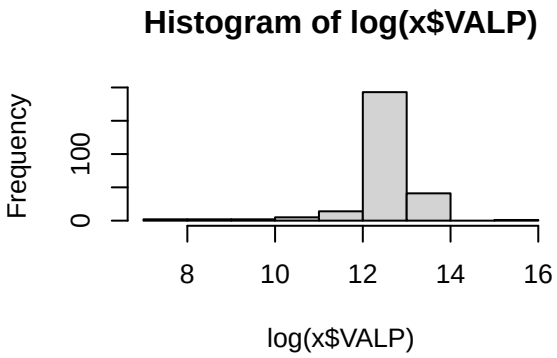

Histogram of log(x$VALP)
Frequency
0
150
6
8
10
12
14
16
log(x$VALP)

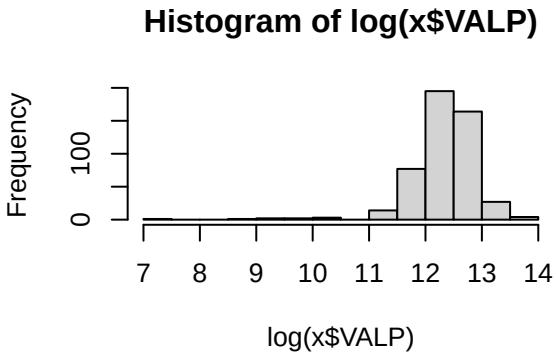

Histogram of log(x$VALP)
Frequency
0
40
7
8
9
10
11
12
13
14
log(x$VALP)

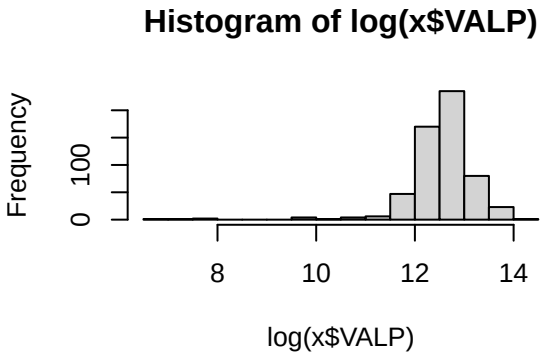

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
log(x$VALP)

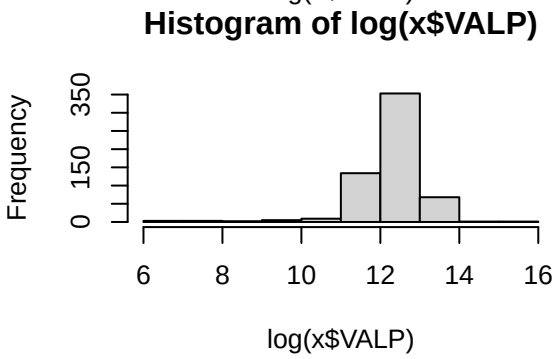

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

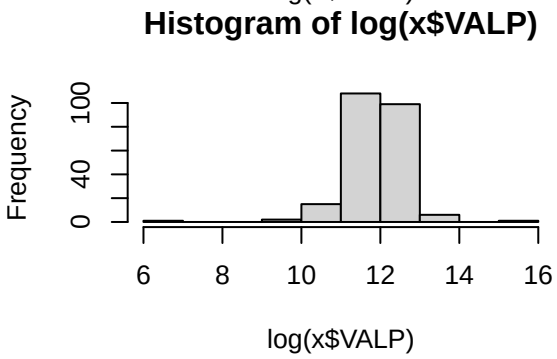

Histogram of log(x$VALP)
Frequency
0
40
100
8
10
12
14
16
log(x$VALP)

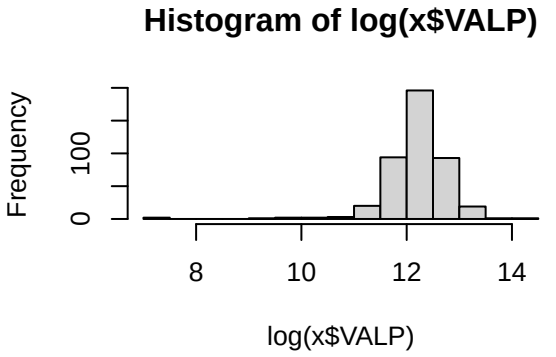

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
16
log(x$VALP)

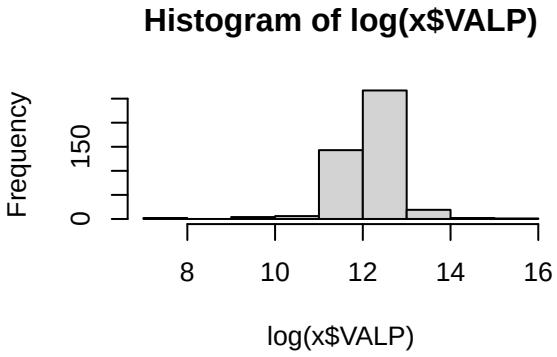

Histogram of log(x$VALP)
Frequency
0
100
7
8
9
10
11
12
13
14
log(x$VALP)

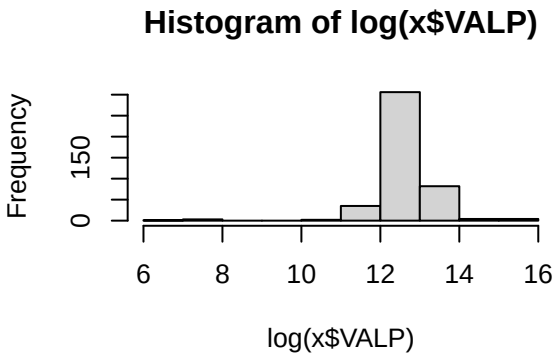

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
log(x$VALP)

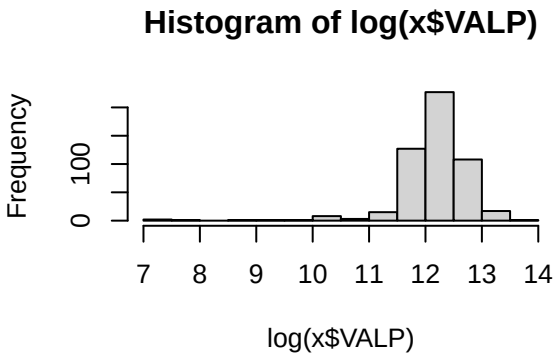

Histogram of log(x$VALP)
Frequency
0
150
350
6
8
10
12
14
16
log(x$VALP)

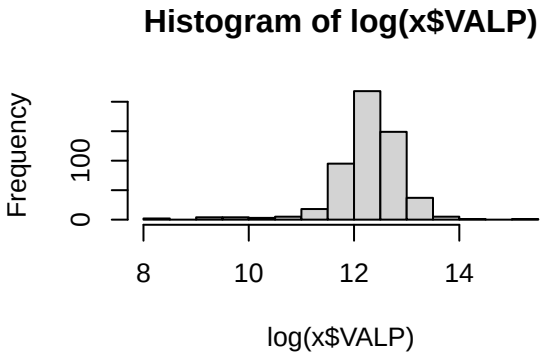

Histogram of log(x$VALP)
Frequency
0
40
100
6
8
10
12
14
16
log(x$VALP)

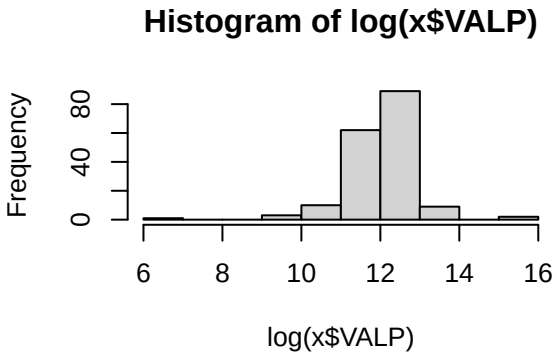

Histogram of log(x$VALP)
Frequency
0
100
8
10
12
14
log(x$VALP)

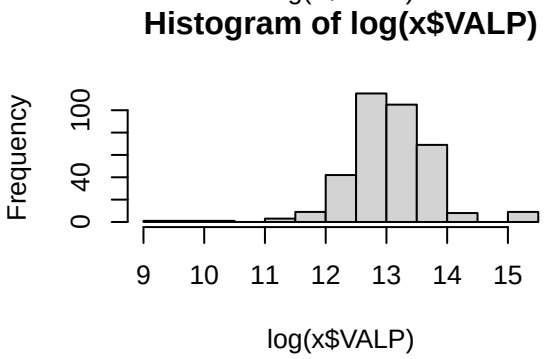

Histogram of log(x$VALP)
Frequency
0
150
8
10
12
14
16
log(x$VALP)

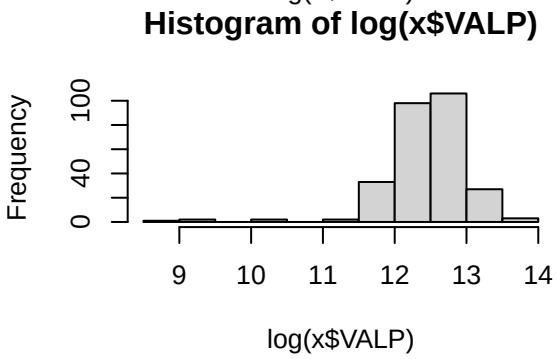


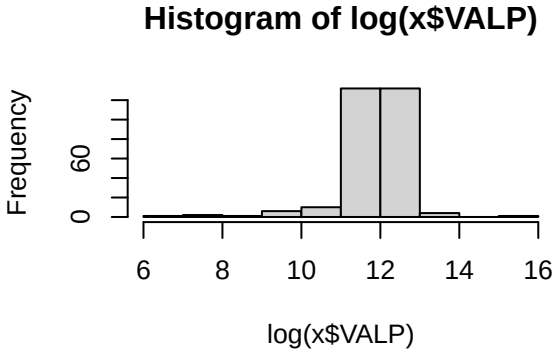


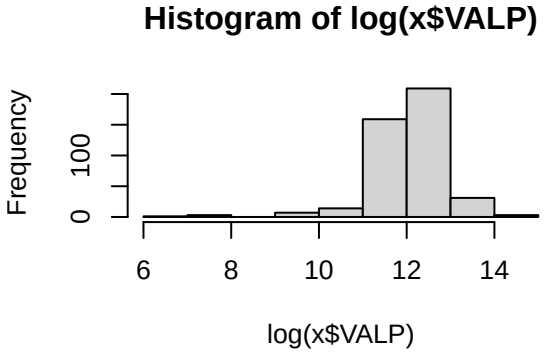










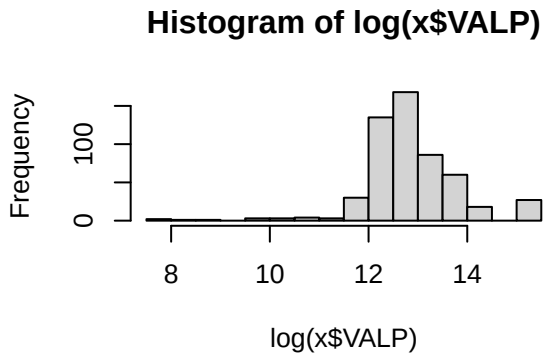



```
##          101          102          301          302          303
## breaks   integer,10   integer,10   integer,10   integer,10   integer,9
## counts   integer,9    integer,9    integer,9    integer,9    integer,8
## density  numeric,9    numeric,9    numeric,9    numeric,9    numeric,8
```

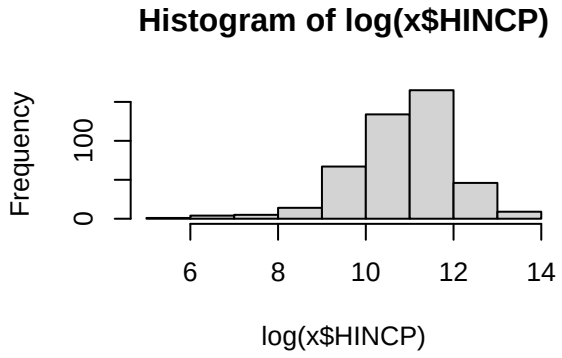
```
## mids     numeric,9     numeric,9     numeric,9     numeric,9     numeric,8
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          304           305           306           307           308
## breaks   integer,11    integer,10    integer,10    numeric,15    integer,11
## counts   integer,10    integer,9     integer,9     integer,14    integer,10
## density  numeric,10    numeric,9     numeric,9     numeric,14    numeric,10
## mids     numeric,10    numeric,9     numeric,9     numeric,14    numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          400           501           502           503           601
## breaks   integer,8     integer,10    integer,9     numeric,10    integer,11
## counts   integer,7     integer,9     integer,8     integer,9     integer,10
## density  numeric,7     numeric,9     numeric,8     numeric,9     numeric,10
## mids     numeric,7     numeric,9     numeric,8     numeric,9     numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          602           701           702           703           800
## breaks   integer,10    integer,10    integer,10    integer,11    integer,11
## counts   integer,9     integer,9     integer,9     integer,10    integer,10
## density  numeric,9     numeric,9     numeric,9     numeric,10    numeric,10
## mids     numeric,9     numeric,9     numeric,9     numeric,10    numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          901           902           903           904           905
## breaks   integer,10    numeric,14    integer,10    integer,10    integer,11
## counts   integer,9     integer,13    integer,9     integer,9     integer,10
## density  numeric,9     numeric,13    numeric,9     numeric,9     numeric,10
## mids     numeric,9     numeric,13    numeric,9     numeric,9     numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          906           907           1001          1002          1003
## breaks   numeric,12    integer,11    integer,10    numeric,11    numeric,14
## counts   integer,11    integer,10    integer,9     integer,10    integer,13
## density  numeric,11    numeric,10    numeric,9     numeric,10    numeric,13
## mids     numeric,11    numeric,10    numeric,9     numeric,10    numeric,13
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          1101          1102          1103          1104          1105
## breaks   integer,11    numeric,15    integer,11    integer,10    integer,10
## counts   integer,10    integer,14    integer,10    integer,9     integer,9
## density  numeric,10    numeric,14    numeric,10    numeric,9     numeric,9
## mids     numeric,10    numeric,14    numeric,10    numeric,9     numeric,9
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          1106          1201          1202          1203          1204
## breaks   numeric,15    integer,10    integer,11    numeric,15    numeric,16
## counts   integer,14    integer,9     integer,10    integer,14    integer,15
## density  numeric,14    numeric,9     numeric,10    numeric,14    numeric,15
## mids     numeric,14    numeric,9     numeric,10    numeric,14    numeric,15
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          1205          1301          1302          1401          1402
## breaks   numeric,12    integer,11    integer,10    integer,9     integer,11
```

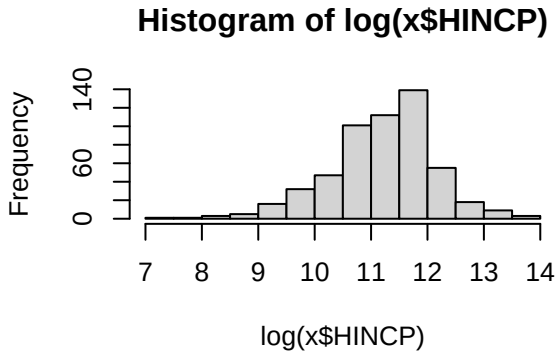
```
## counts   integer,11    integer,10    integer,9     integer,8     integer,10
## density  numeric,11    numeric,10    numeric,9     numeric,8     numeric,10
## mids     numeric,11    numeric,10    numeric,9     numeric,8     numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          1403          1404          1501          1502          1503
## breaks   integer,10    integer,10    integer,10    integer,11    integer,9
## counts   integer,9     integer,9     integer,9     integer,10    integer,8
## density  numeric,9     numeric,9     numeric,9     numeric,10    numeric,8
## mids     numeric,9     numeric,9     numeric,9     numeric,10    numeric,8
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          1504          1600          1700          1800          1901
## breaks   integer,11    numeric,14    integer,11    integer,8     numeric,16
## counts   integer,10    integer,13    integer,10    integer,7     integer,15
## density  numeric,10    numeric,13    numeric,10    numeric,7     numeric,15
## mids     numeric,10    numeric,13    numeric,10    numeric,7     numeric,15
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          1902          1903          1904          2001          2002
## breaks   integer,10    integer,10    integer,10    numeric,15    numeric,17
## counts   integer,9     integer,9     integer,9     integer,14    integer,16
## density  numeric,9     numeric,9     numeric,9     numeric,14    numeric,16
## mids     numeric,9     numeric,9     numeric,9     numeric,14    numeric,16
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          2003          2101          2102          2103          2104
## breaks   integer,11    integer,11    numeric,16    integer,10    integer,11
## counts   integer,10    integer,10    integer,15    integer,9     integer,10
## density  numeric,10    numeric,10    numeric,15    numeric,9     numeric,10
## mids     numeric,10    numeric,10    numeric,15    numeric,9     numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          2201          2202          2301          2302          2303
## breaks   numeric,15    numeric,16    integer,11    numeric,14    numeric,12
## counts   integer,14    integer,15    integer,10    integer,13    integer,11
## density  numeric,14    numeric,15    numeric,10    numeric,13    numeric,11
## mids     numeric,14    numeric,15    numeric,10    numeric,13    numeric,11
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          2400          2500          2600
## breaks   integer,11    integer,10    numeric,17
## counts   integer,10    integer,9     integer,16
## density  numeric,10    numeric,9     numeric,16
## mids     numeric,10    numeric,9     numeric,16
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE
```

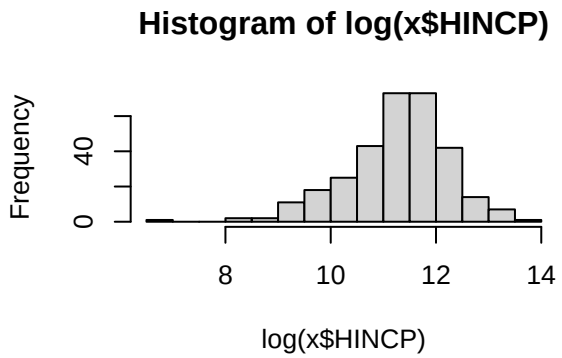
```
par(mfrow = c(2, 2))
```

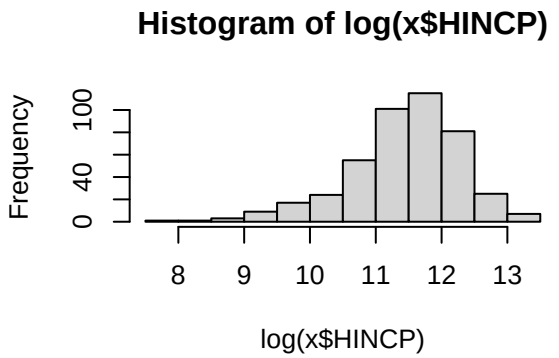


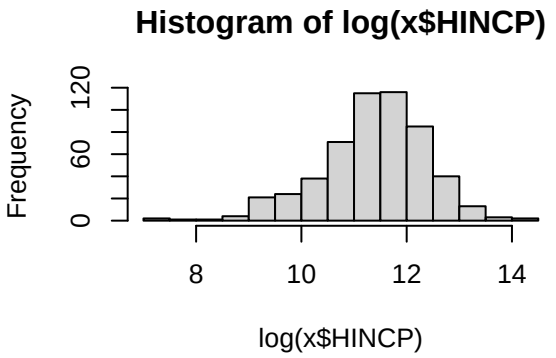

```
sapply(split(pumsnj, pumsnj$PUMA), function(x) {
    hist(log(x$HINCP))
})
```

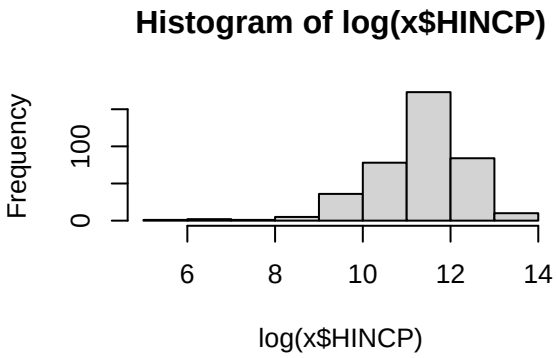


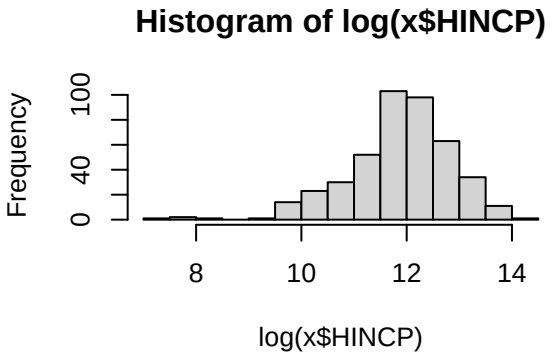


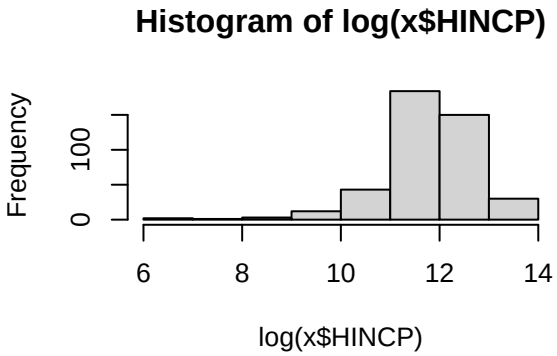




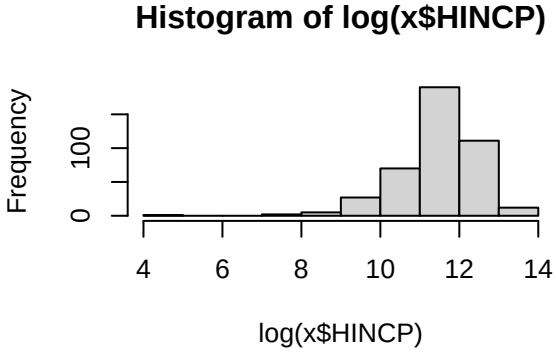


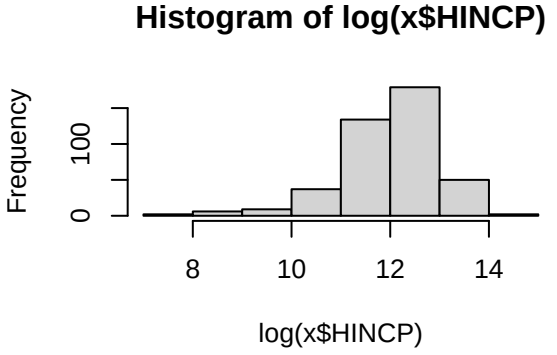


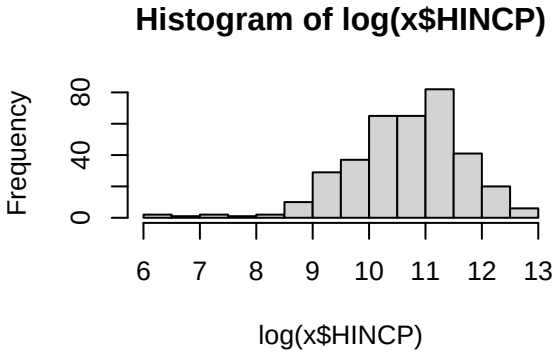


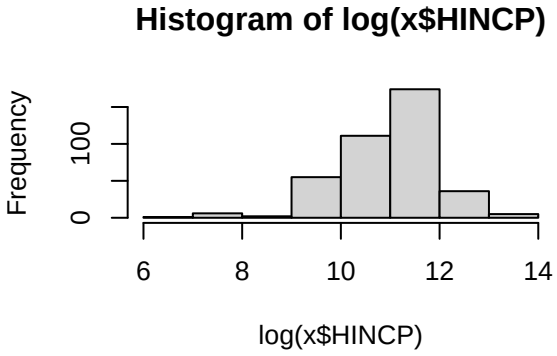


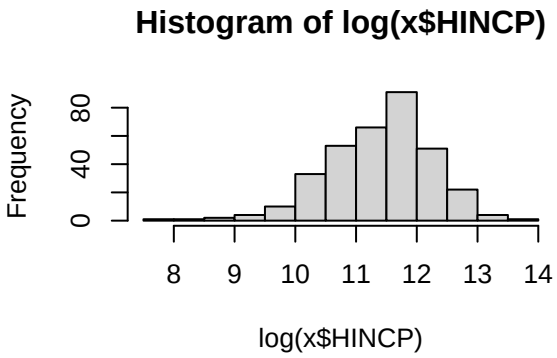

```
## Warning in log(x$HINCP): NaNs produced
```

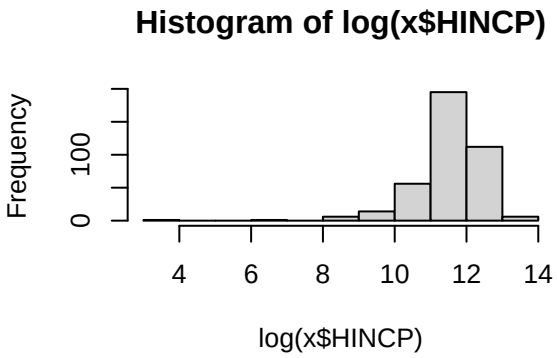


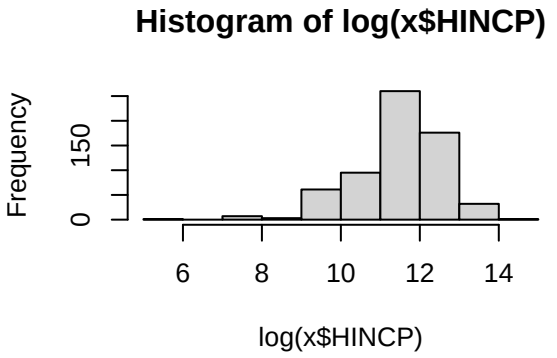


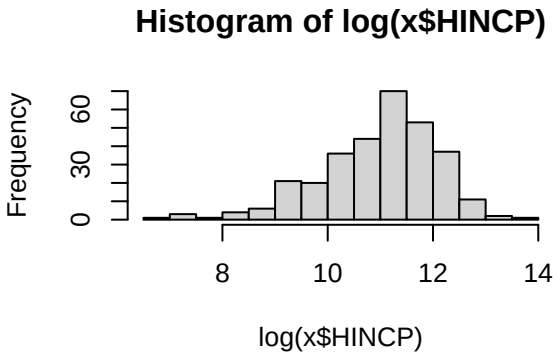




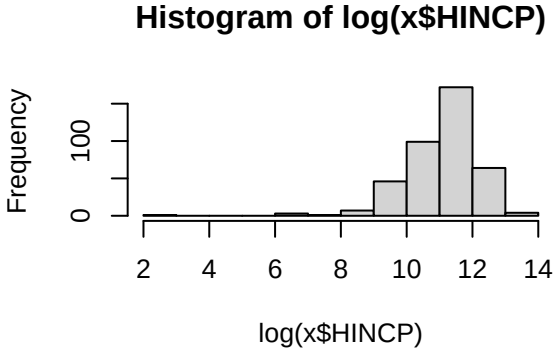


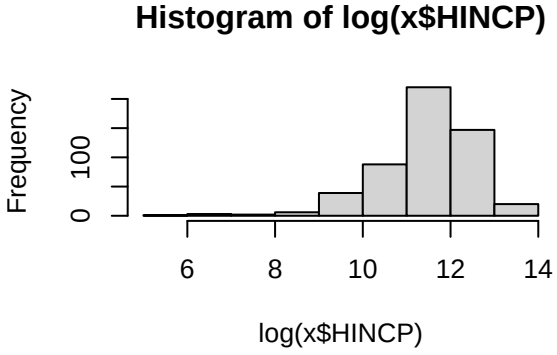


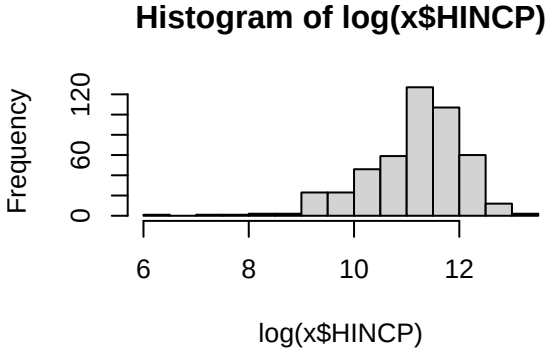


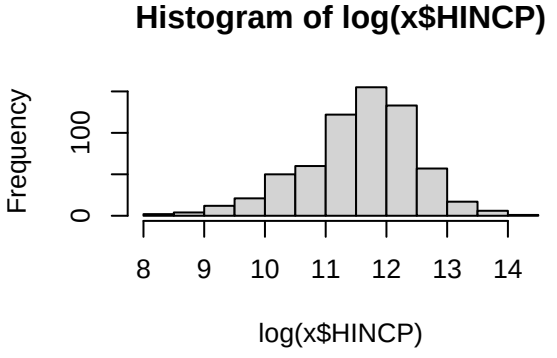


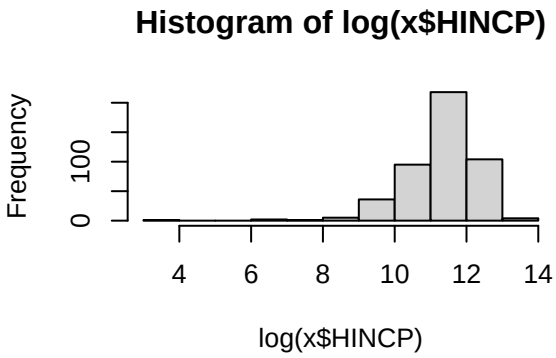

```
## Warning in log(x$HINCP): NaNs produced
```

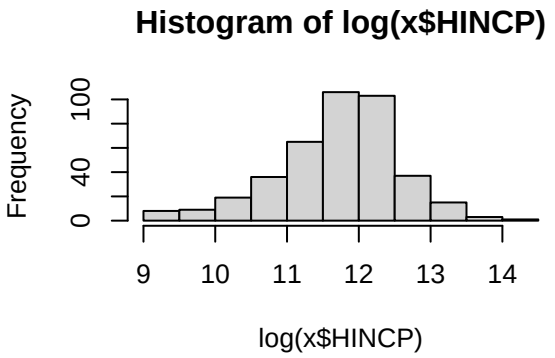


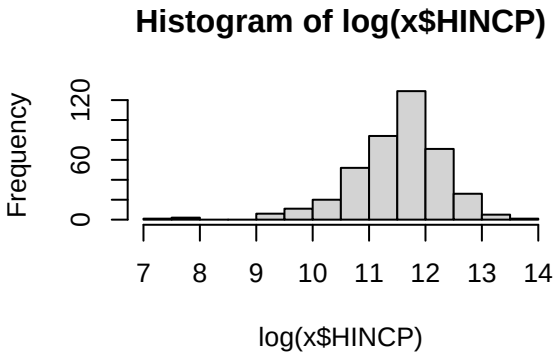


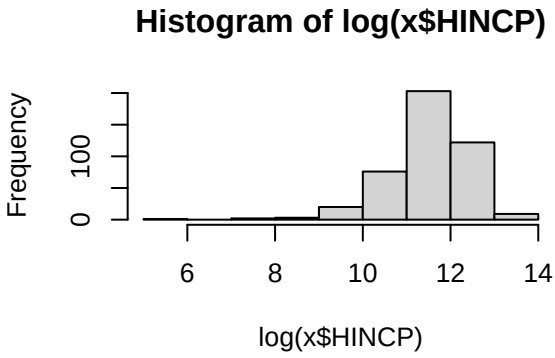




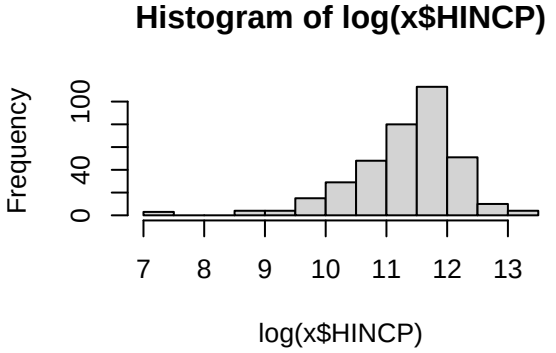


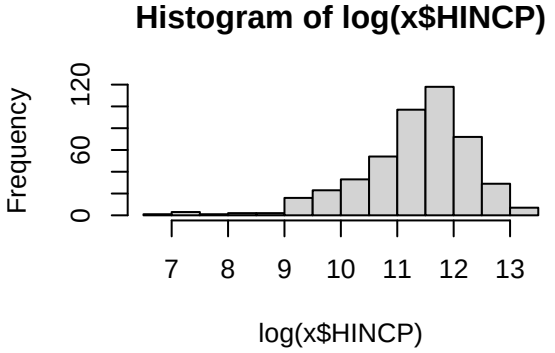


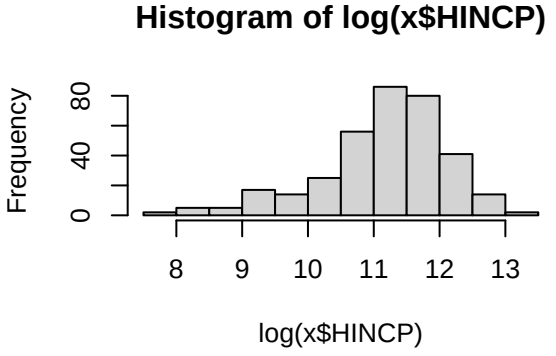


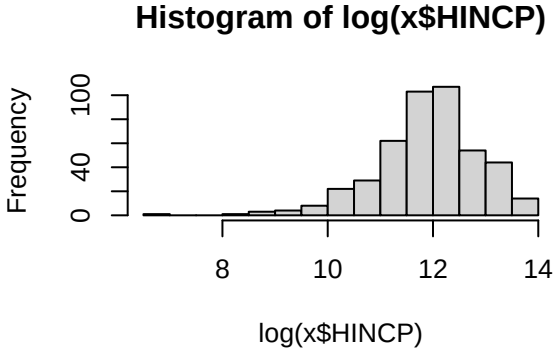



```
## Warning in log(x$HINCP): NaNs produced
```

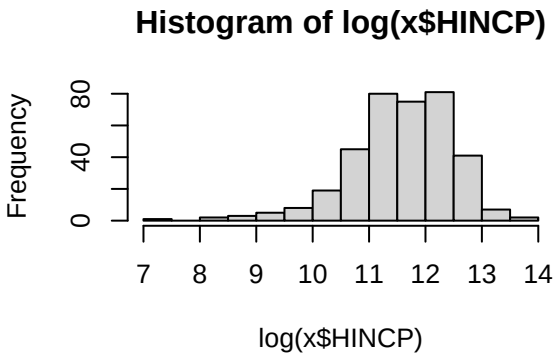


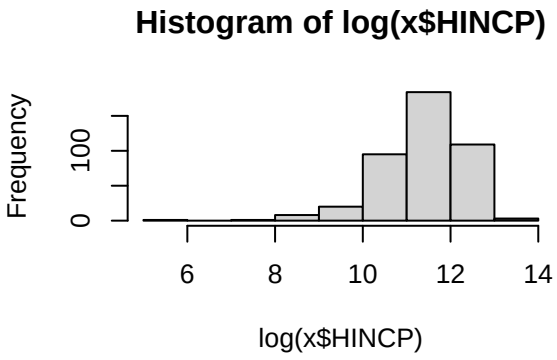


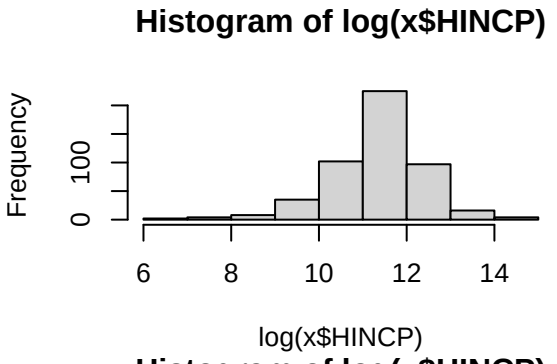


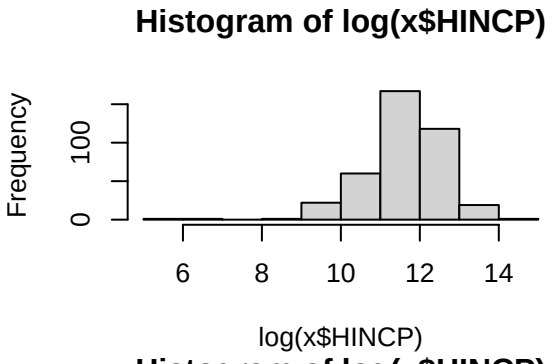


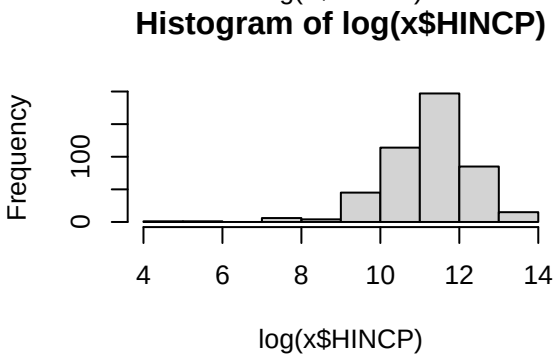

```
## Warning in log(x$HINCP): NaNs produced
```

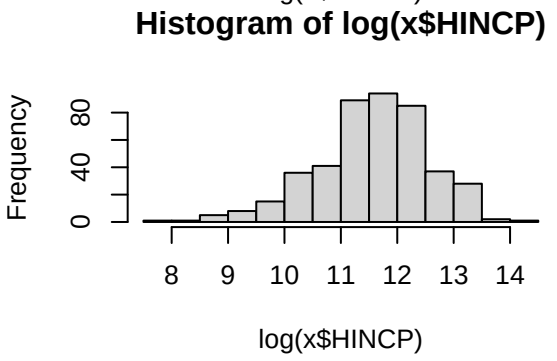


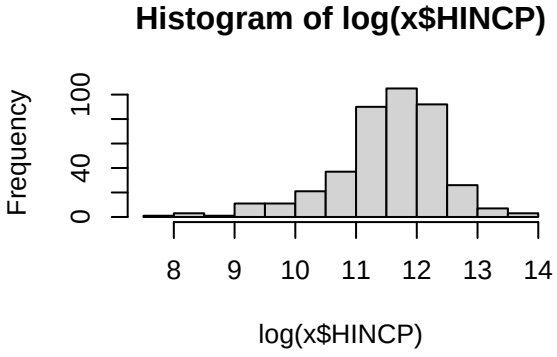


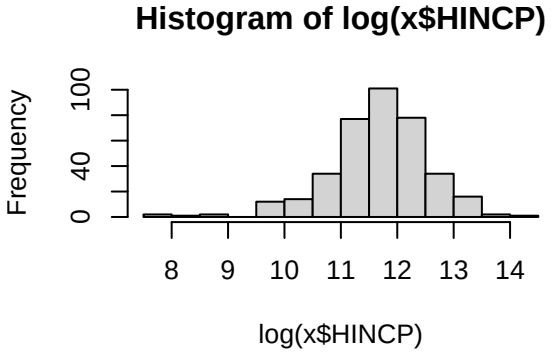




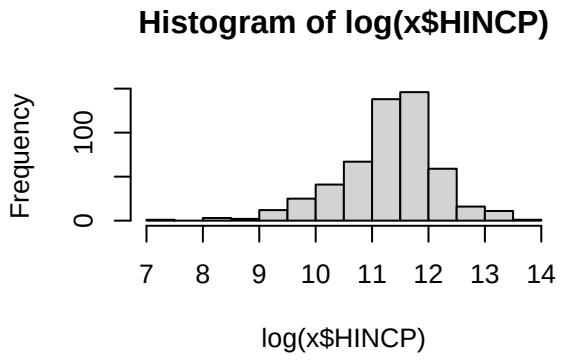


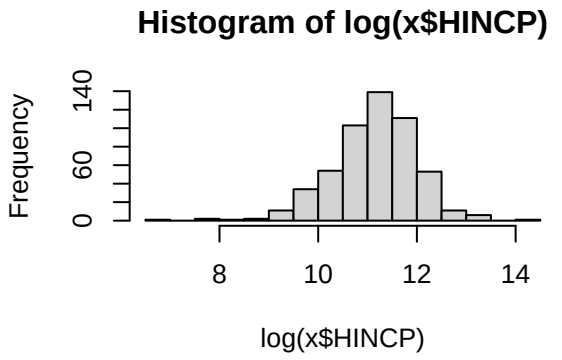


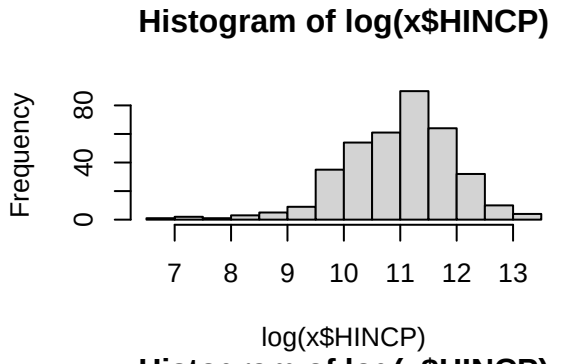


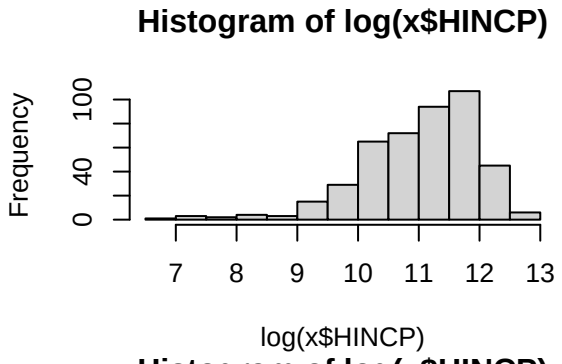


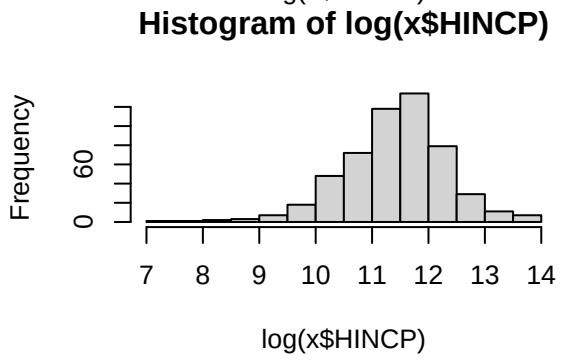

```
## Warning in log(x$HINCP): NaNs produced

## Warning in log(x$HINCP): NaNs produced
```

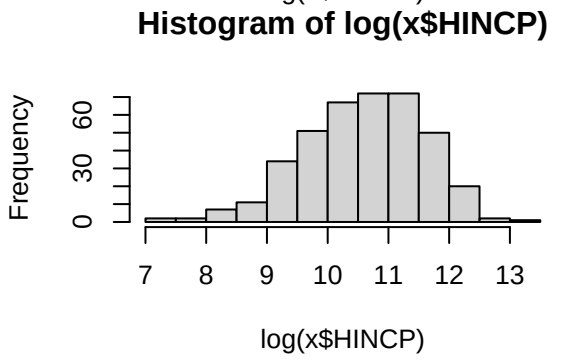

Histogram of log(x$HINCP)
Frequency
0
100
7
8
9
10
11
12
13
14
log(x$HINCP)

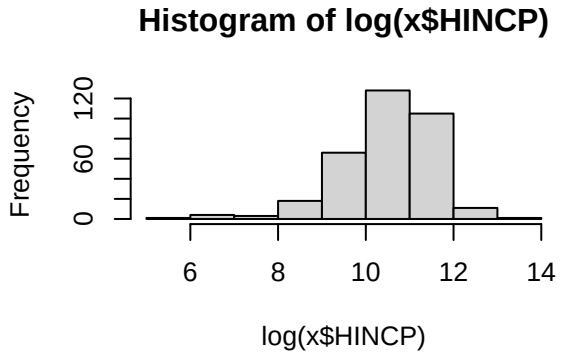

Histogram of log(x$HINCP)
Frequency
0
60
140
8
10
12
14
log(x$HINCP)

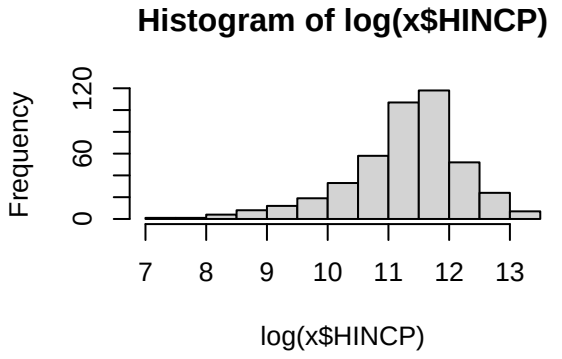

Histogram of log(x$HINCP)
Frequency
0
40
80
7
8
9
10
11
12
13
log(x$HINCP)

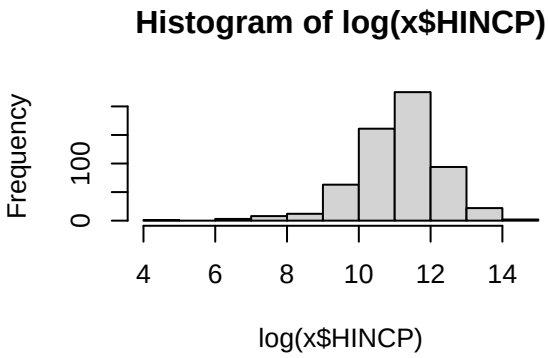

Histogram of log(x$HINCP)
Frequency
0
40
100
7
8
9
10
11
12
13
log(x$HINCP)

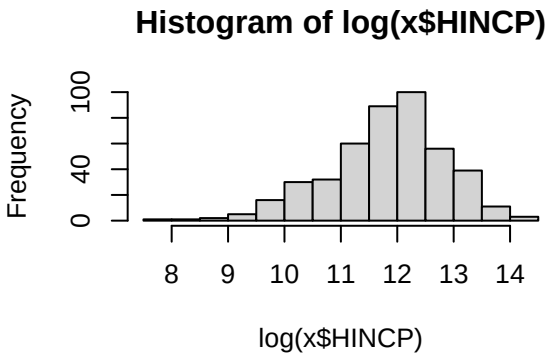

Histogram of log(x$HINCP)
Frequency
0
60
7
8
9
10
11
12
13
14
log(x$HINCP)

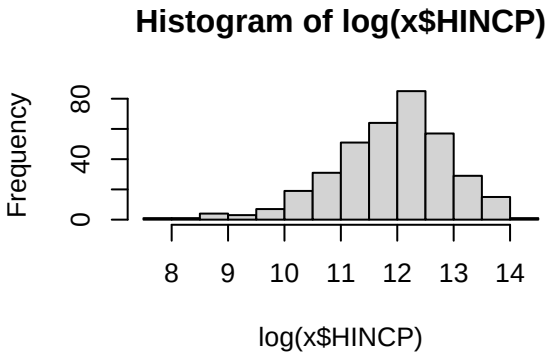

Histogram of log(x$HINCP)
Frequency
0
30
60
7
8
9
10
11
12
13
log(x$HINCP)

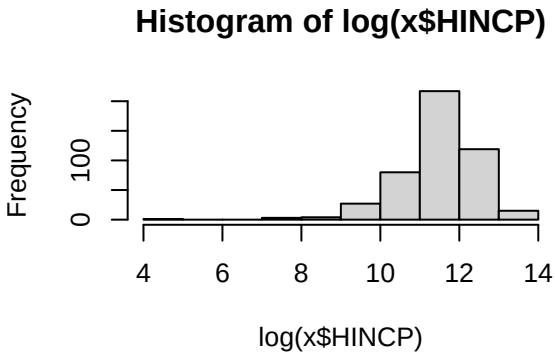

Histogram of log(x$HINCP)
Frequency
0
60
120
6
8
10
12
14
log(x$HINCP)


Histogram of log(x$HINCP)
Frequency
0
60
120
7
8
9
10
11
12
13
log(x$HINCP)

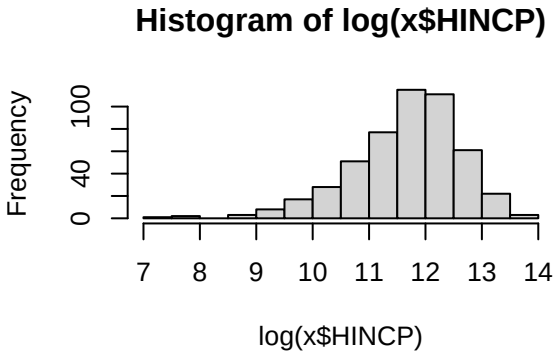


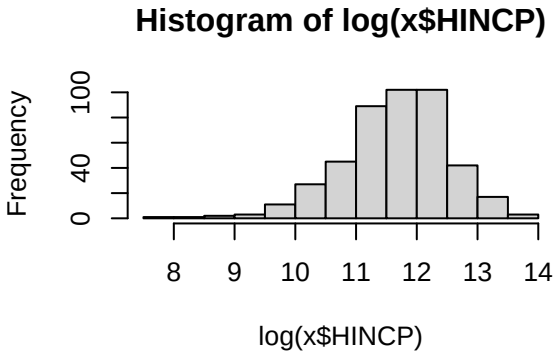


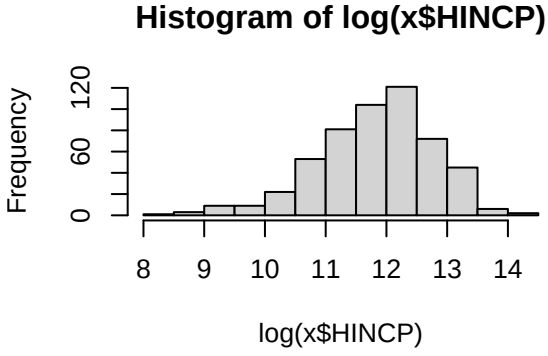


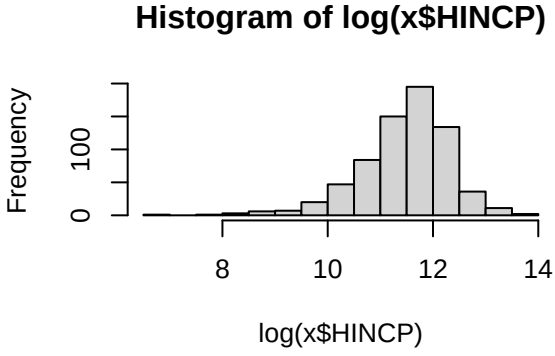



```
## Warning in log(x$HINCP): NaNs produced
```

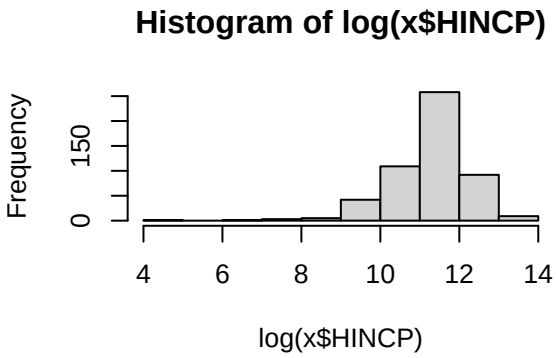


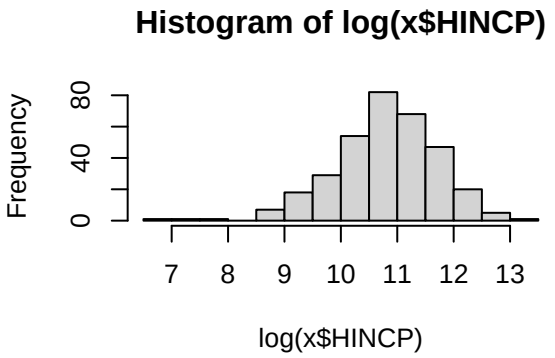


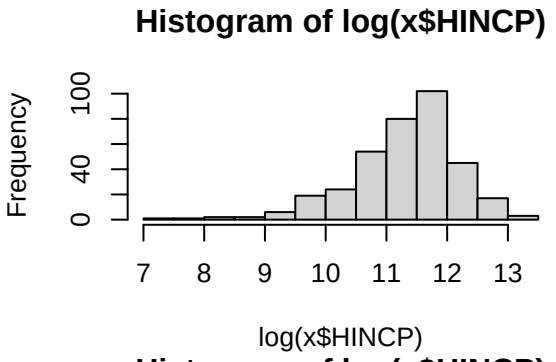


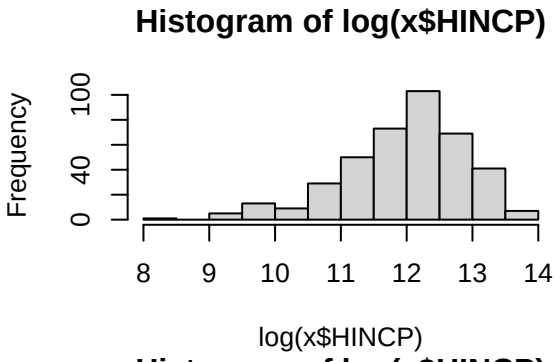


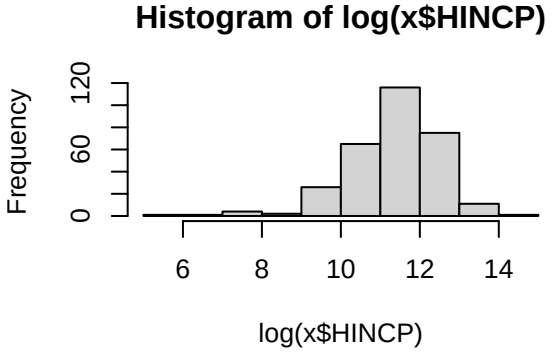

```
## Warning in log(x$HINCP): NaNs produced
```

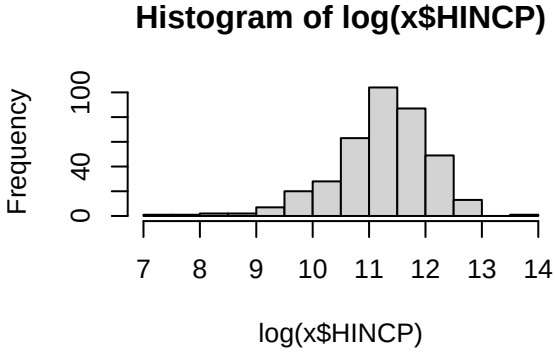

Histogram of log(x$HINCP)
Frequency
0
150
4
6
8
10
12
14
log(x$HINCP)

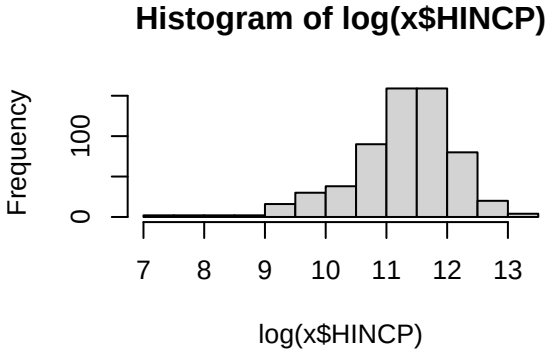

Histogram of log(x$HINCP)
Frequency
0
40
80
7
8
9
10
11
12
13
log(x$HINCP)

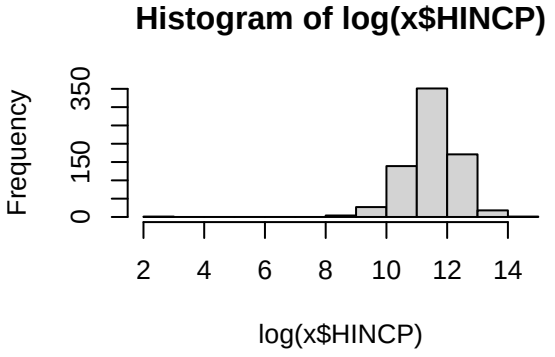

Histogram of log(x$HINCP)
Frequency
0
40
100
7
8
9
10
11
12
13
log(x$HINCP)

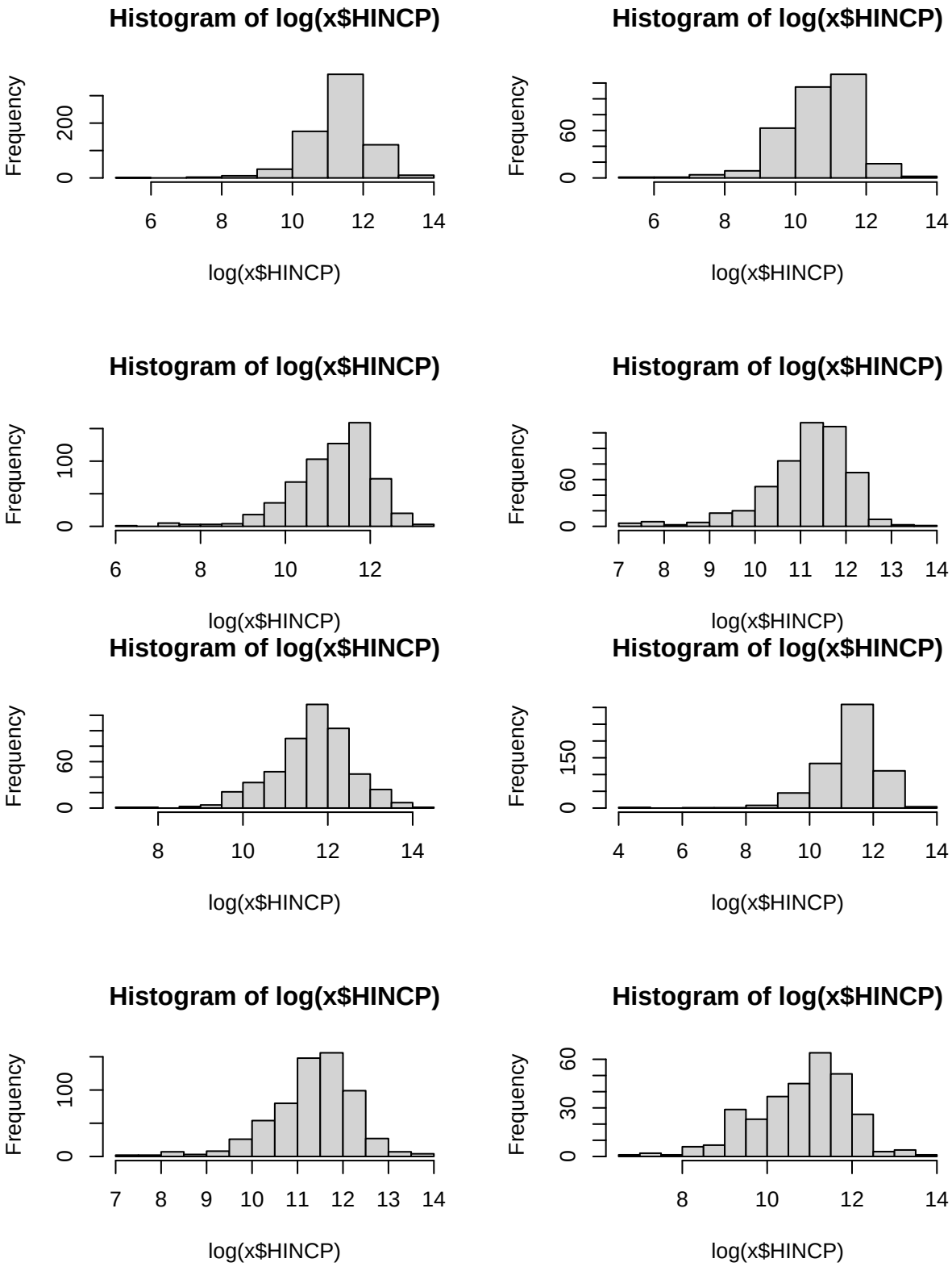

Histogram of log(x$HINCP)
Frequency
0
40
100
8
9
10
11
12
13
14
log(x$HINCP)


Histogram of log(x$HINCP)
Frequency
0
60
120
6
8
10
12
14
log(x$HINCP)

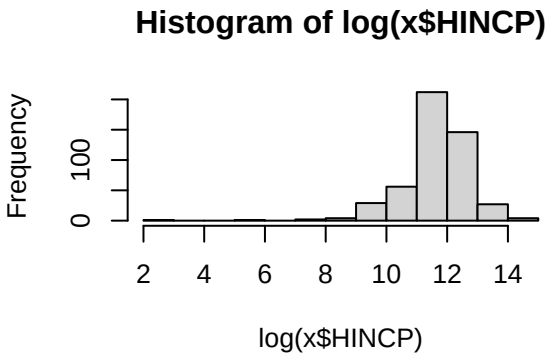

Histogram of log(x$HINCP)
Frequency
0
40
100
7
8
9
10
11
12
13
14
log(x$HINCP)

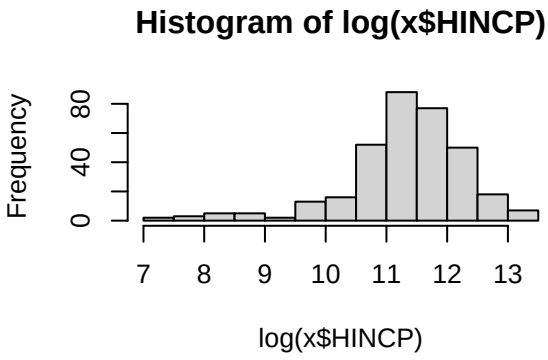

Histogram of log(x$HINCP)
Frequency
0
100
7
8
9
10
11
12
13
log(x$HINCP)

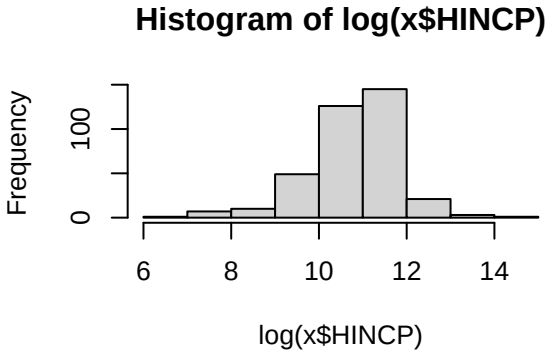

Histogram of log(x$HINCP)
Frequency
0
150
350
2
4
6
8
10
12
14
log(x$HINCP)

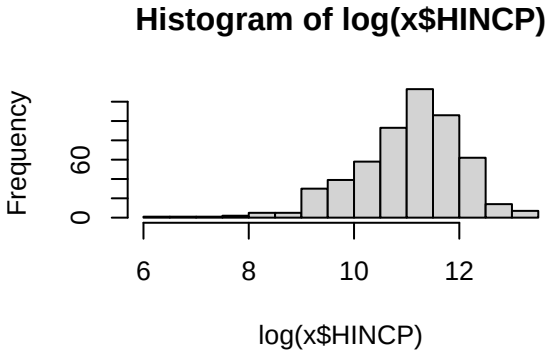



```
## Warning in log(x$HINCP): NaNs produced
```







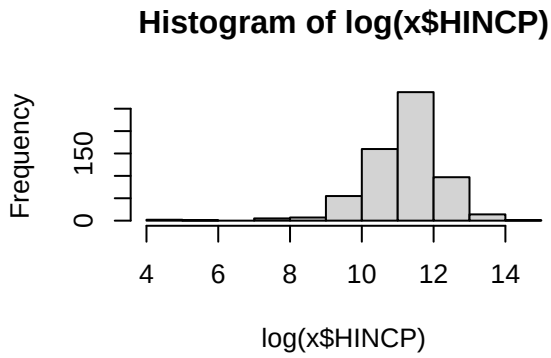



```
##          101             102             301             302
## breaks   integer,10      numeric,15      numeric,16      numeric,13
## counts   integer,9       integer,14      integer,15      integer,12
## density  numeric,9       numeric,14      numeric,15      numeric,12
## mids     numeric,9       numeric,14      numeric,15      numeric,12
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE            TRUE            TRUE            TRUE
##          303             304             305             306
## breaks   numeric,16      integer,10      numeric,16      integer,9
## counts   integer,15      integer,9       integer,15      integer,8
## density  numeric,15      numeric,9       numeric,15      numeric,8
## mids     numeric,15      numeric,9       numeric,15      numeric,8
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE            TRUE            TRUE            TRUE
##          307             308             400             501
## breaks   integer,11      integer,9       numeric,15      integer,9
## counts   integer,10      integer,8       integer,14      integer,8
## density  numeric,10      numeric,8       numeric,14      numeric,8
## mids     numeric,10      numeric,8       numeric,14      numeric,8
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE            TRUE            TRUE            TRUE
##          502             503             601             602
## breaks   numeric,14      integer,12      integer,11      numeric,16
## counts   integer,13      integer,11      integer,10      integer,15
## density  numeric,13      numeric,11      numeric,10      numeric,15
## mids     numeric,13      numeric,11      numeric,10      numeric,15
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE            TRUE            TRUE            TRUE
```

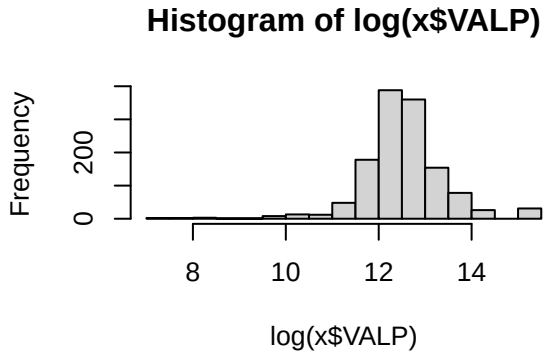
```
##          701            702            703            800
## breaks   integer,13     integer,10     numeric,16     numeric,14
## counts   integer,12     integer,9      integer,15     integer,13
## density  numeric,12     numeric,9      numeric,15     numeric,13
## mids     numeric,12     numeric,9      numeric,15     numeric,13
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          901            902            903            904
## breaks   integer,12     numeric,12     numeric,15     integer,10
## counts   integer,11     integer,11     integer,14     integer,9
## density  numeric,11     numeric,11     numeric,14     numeric,9
## mids     numeric,11     numeric,11     numeric,14     numeric,9
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          905            906            907            1001
## breaks   numeric,14     numeric,15     numeric,13     numeric,16
## counts   integer,13     integer,14     integer,12     integer,15
## density  numeric,13     numeric,14     numeric,12     numeric,15
## mids     numeric,13     numeric,14     numeric,12     numeric,15
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1002           1003           1101           1102
## breaks   numeric,15     integer,10     integer,10     integer,11
## counts   integer,14     integer,9      integer,9      integer,10
## density  numeric,14     numeric,9      numeric,9      numeric,10
## mids     numeric,14     numeric,9      numeric,9      numeric,10
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1103           1104           1105           1106
## breaks   integer,11     numeric,15     numeric,14     numeric,15
## counts   integer,10     integer,14     integer,13     integer,14
## density  numeric,10     numeric,14     numeric,13     numeric,14
## mids     numeric,10     numeric,14     numeric,13     numeric,14
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1201           1202           1203           1204
## breaks   numeric,15     numeric,17     numeric,15     numeric,14
## counts   integer,14     integer,16     integer,14     integer,13
## density  numeric,14     numeric,16     numeric,14     numeric,13
## mids     numeric,14     numeric,16     numeric,14     numeric,13
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1205           1301           1302           1401
## breaks   numeric,15     numeric,14     integer,10     numeric,14
## counts   integer,14     integer,13     integer,9      integer,13
## density  numeric,14     numeric,13     numeric,9      numeric,13
## mids     numeric,14     numeric,13     numeric,9      numeric,13
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1402           1403           1404           1501
## breaks   integer,12     numeric,15     numeric,15     integer,11
## counts   integer,11     integer,14     integer,14     integer,10
## density  numeric,11     numeric,14     numeric,14     numeric,10
## mids     numeric,11     numeric,14     numeric,14     numeric,10
```

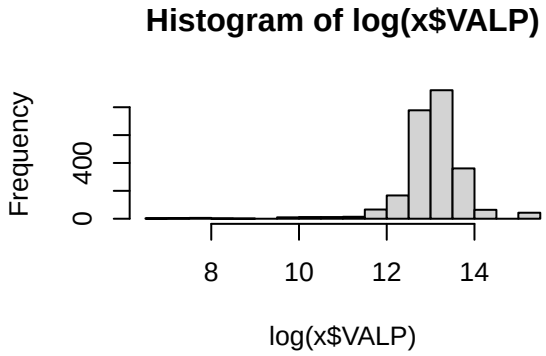
```
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1502           1503           1504           1600
## breaks   numeric,15     numeric,14     numeric,14     numeric,16
## counts   integer,14     integer,13     integer,13     integer,15
## density  numeric,14     numeric,13     numeric,13     numeric,15
## mids     numeric,14     numeric,13     numeric,13     numeric,15
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1700           1800           1901           1902
## breaks   integer,11     numeric,15     numeric,14     numeric,13
## counts   integer,10     integer,14     integer,13     integer,12
## density  numeric,10     numeric,14     numeric,13     numeric,12
## mids     numeric,10     numeric,14     numeric,13     numeric,12
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          1903           1904           2001           2002
## breaks   integer,11     numeric,15     numeric,14     integer,14
## counts   integer,10     integer,14     integer,13     integer,13
## density  numeric,10     numeric,14     numeric,13     numeric,13
## mids     numeric,10     numeric,14     numeric,13     numeric,13
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          2003           2101           2102           2103
## breaks   integer,10     integer,10     numeric,16     numeric,15
## counts   integer,9      integer,9      integer,15     integer,14
## density  numeric,9      numeric,9      numeric,15     numeric,14
## mids     numeric,9      numeric,9      numeric,15     numeric,14
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          2104           2201           2202           2301
## breaks   numeric,16     integer,11     numeric,15     numeric,16
## counts   integer,15     integer,10     integer,14     integer,15
## density  numeric,15     numeric,10     numeric,14     numeric,15
## mids     numeric,15     numeric,10     numeric,14     numeric,15
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          2302           2303           2400           2500
## breaks   integer,14     numeric,14     integer,10     numeric,16
## counts   integer,13     integer,13     integer,9      integer,15
## density  numeric,13     numeric,13     numeric,9      numeric,15
## mids     numeric,13     numeric,13     numeric,9      numeric,15
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          2600
## breaks   integer,12
## counts   integer,11
## density  numeric,11
## mids     numeric,11
## xname    "log(x$HINCP)"
## equidist TRUE
```

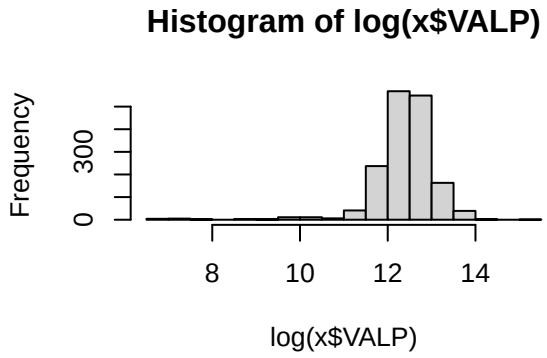
```r
# Within FIPS
par(mfrow = c(2, 2))
```

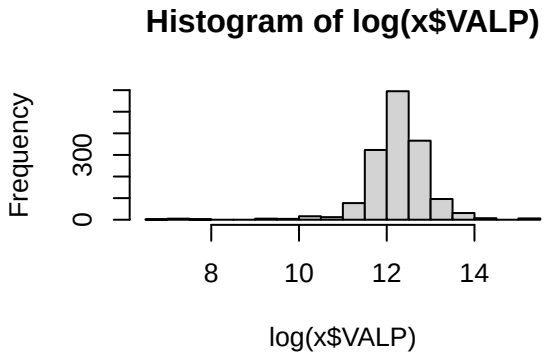


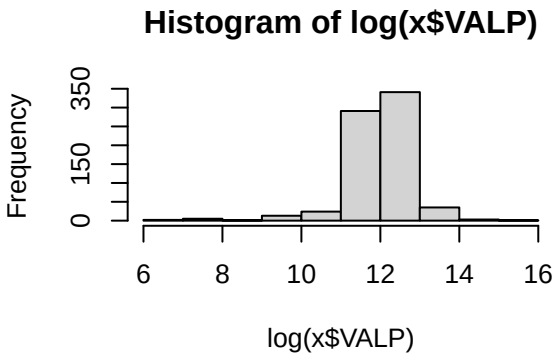

```
sapply(split(pumsnj, pumsnj$fips), function(x) {
    hist(log(x$VALP))
})
```

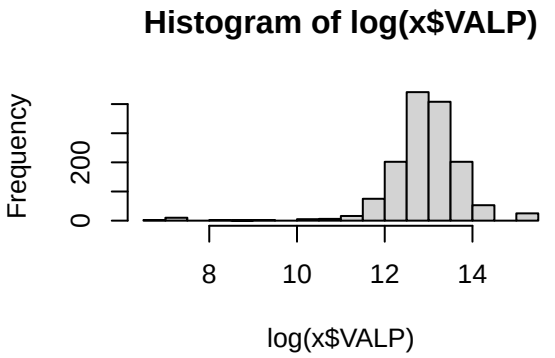


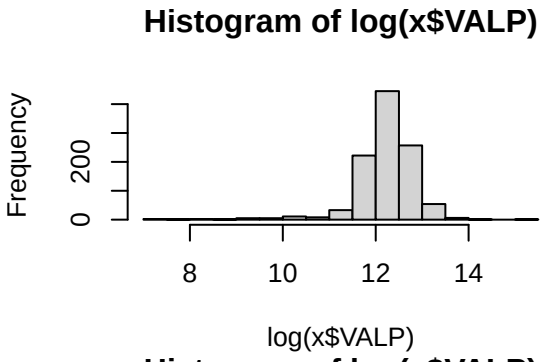


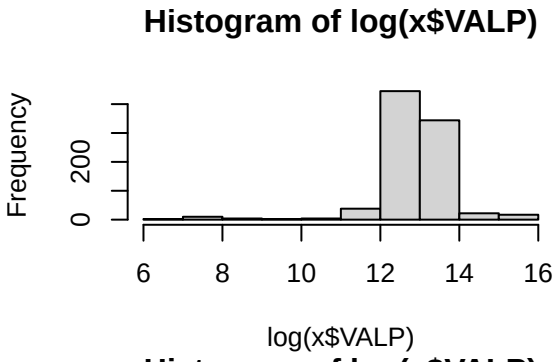


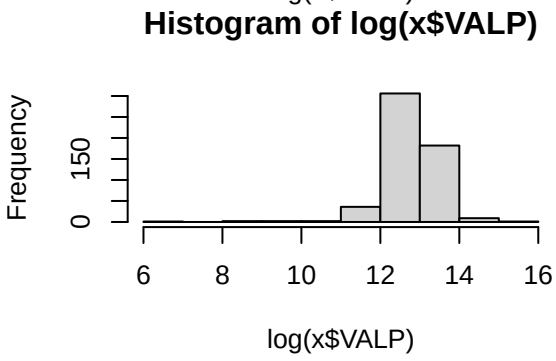


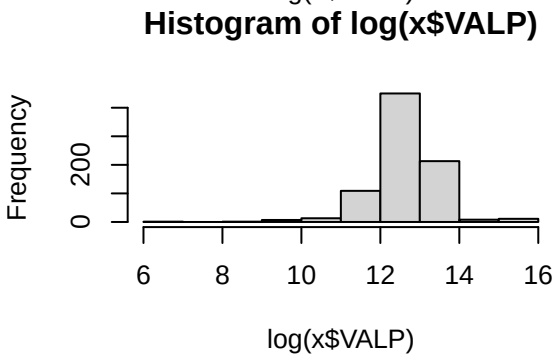

Histogram of log(x$VALP)
Frequency
0
150
350
6
8
10
12
14
16
log(x$VALP)

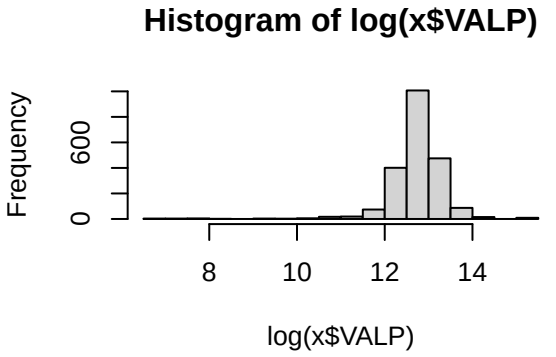

Histogram of log(x$VALP)
Frequency
0
200
8
10
12
14
log(x$VALP)

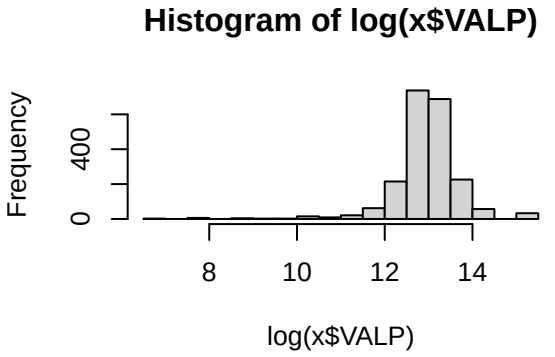

Histogram of log(x$VALP)
Frequency
0
200
8
10
12
14
log(x$VALP)

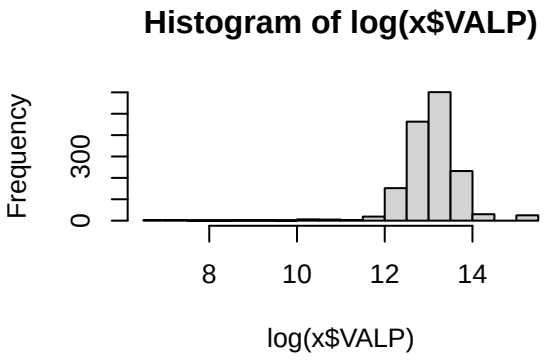

Histogram of log(x$VALP)
Frequency
0
200
6
8
10
12
14
16
log(x$VALP)

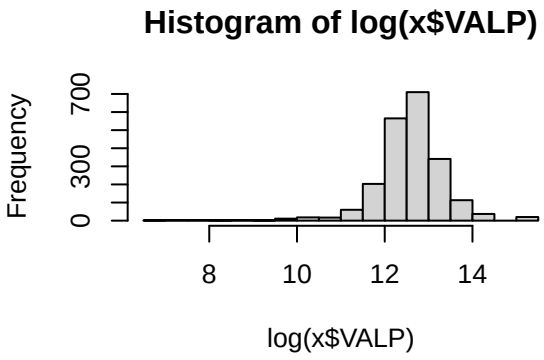

Histogram of log(x$VALP)
Frequency
0
150
6
8
10
12
14
16
log(x$VALP)

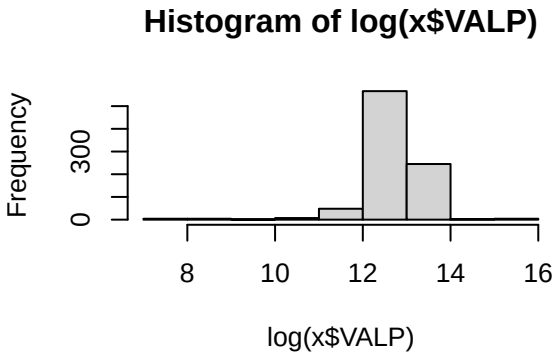

Histogram of log(x$VALP)
Frequency
0
200
6
8
10
12
14
16
log(x$VALP)

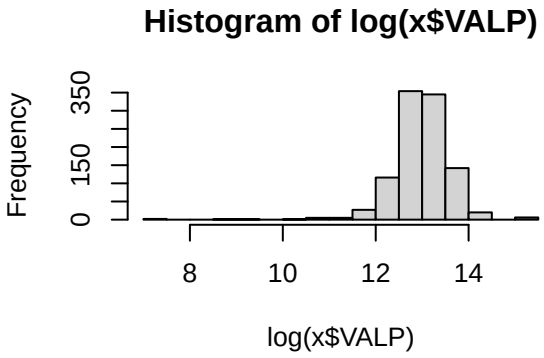

Histogram of log(x$VALP)
Frequency
0
600
8
10
12
14
log(x$VALP)


Histogram of log(x$VALP)
Frequency
0
400
8
10
12
14
log(x$VALP)



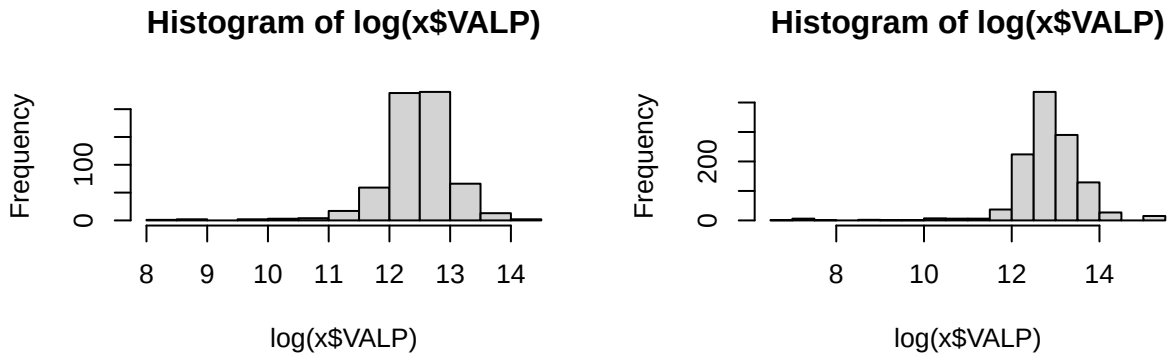


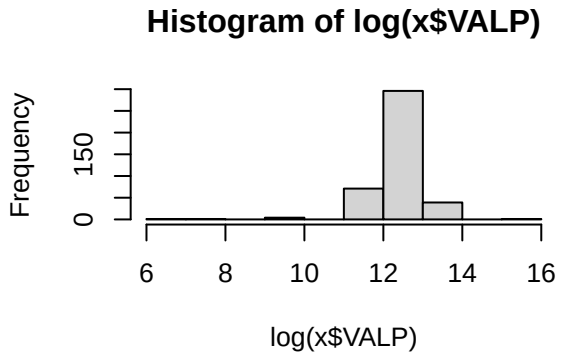





```
##          34001         34003         34005         34007         34011
## breaks   numeric,18    numeric,19    numeric,19    numeric,19    integer,11
## counts   integer,17    integer,18    integer,18    integer,18    integer,10
## density  numeric,17    numeric,18    numeric,18    numeric,18    numeric,10
## mids     numeric,17    numeric,18    numeric,18    numeric,18    numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          34013         34015         34017         34019         34021
## breaks   numeric,19    numeric,18    integer,11    integer,11    integer,11
## counts   integer,18    integer,17    integer,10    integer,10    integer,10
## density  numeric,18    numeric,17    numeric,10    numeric,10    numeric,10
## mids     numeric,18    numeric,17    numeric,10    numeric,10    numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          34023         34025         34027         34029         34031
## breaks   numeric,19    numeric,19    numeric,19    numeric,19    integer,10
## counts   integer,18    integer,18    integer,18    integer,18    integer,9
## density  numeric,18    numeric,18    numeric,18    numeric,18    numeric,9
## mids     numeric,18    numeric,18    numeric,18    numeric,18    numeric,9
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE          TRUE
##          34035         34037         34039         34041
## breaks   numeric,18    numeric,14    numeric,19    integer,11
## counts   integer,17    integer,13    integer,18    integer,10
## density  numeric,17    numeric,13    numeric,18    numeric,10
## mids     numeric,17    numeric,13    numeric,18    numeric,10
## xname    "log(x$VALP)" "log(x$VALP)" "log(x$VALP)" "log(x$VALP)"
## equidist TRUE          TRUE          TRUE          TRUE
```

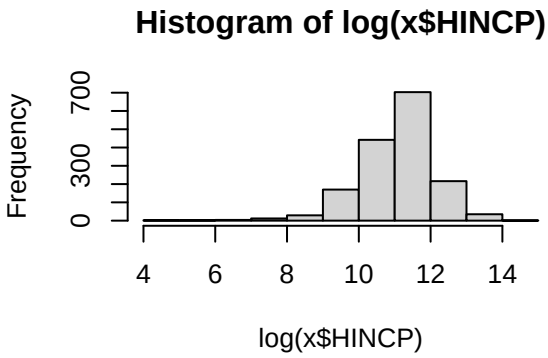
```
par(mfrow = c(2, 2))
```

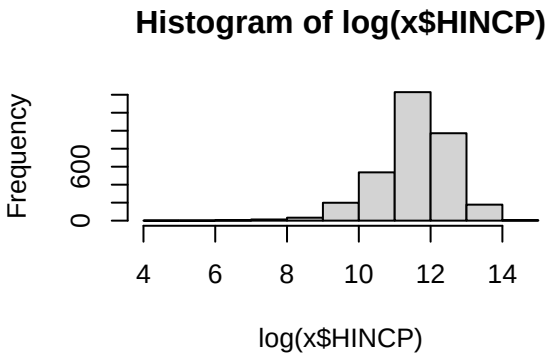


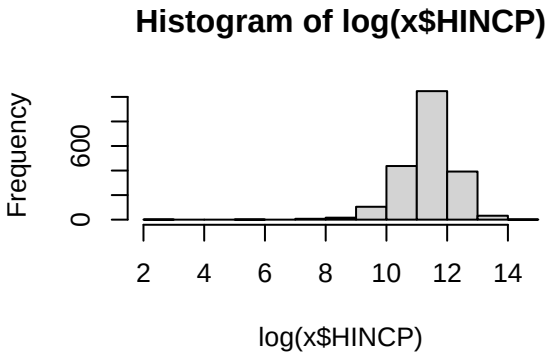


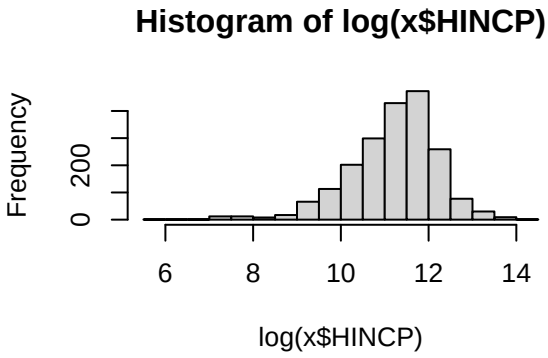

```
sapply(split(pumsnj, pumsnj$fips), function(x) {
    hist(log(x$HINCP))
})
```

```
## Warning in log(x$HINCP): NaNs produced
```

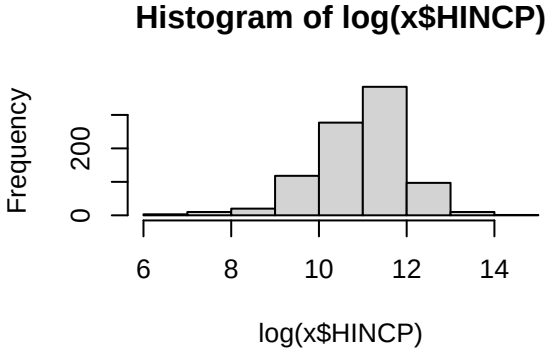


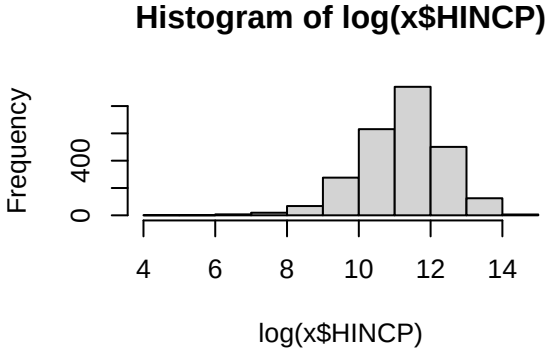


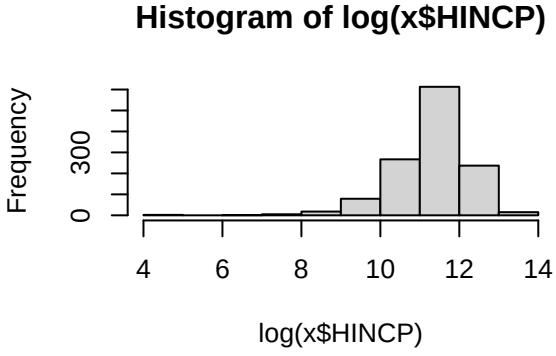


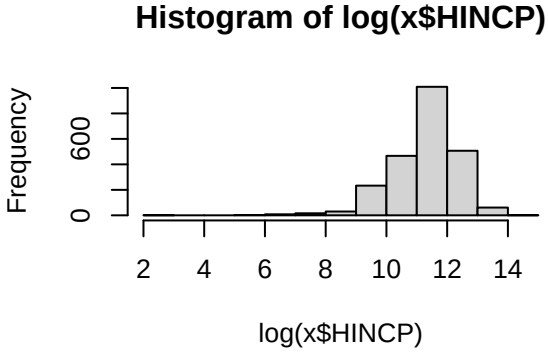



```
## Warning in log(x$HINCP): NaNs produced
```

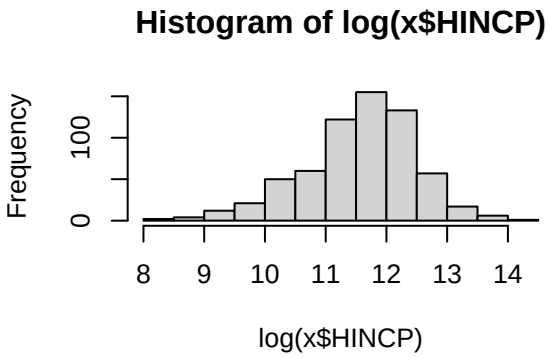


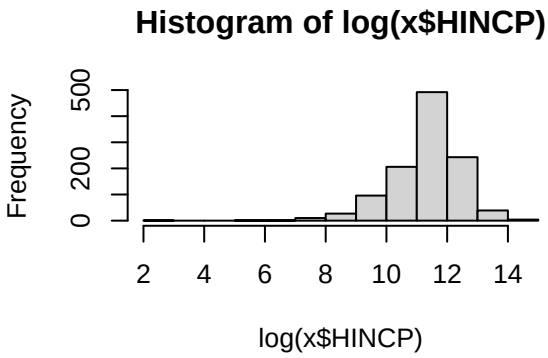


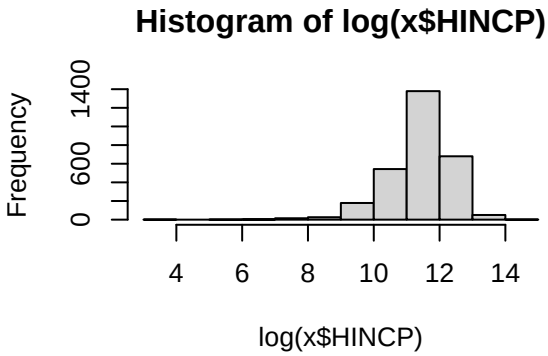


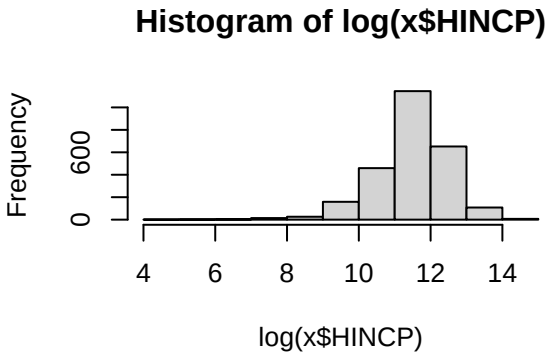



```
## Warning in log(x$HINCP): NaNs produced
```

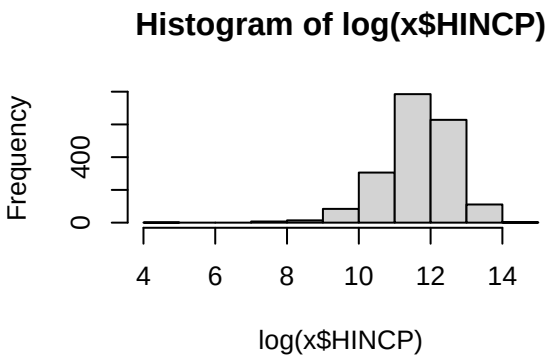


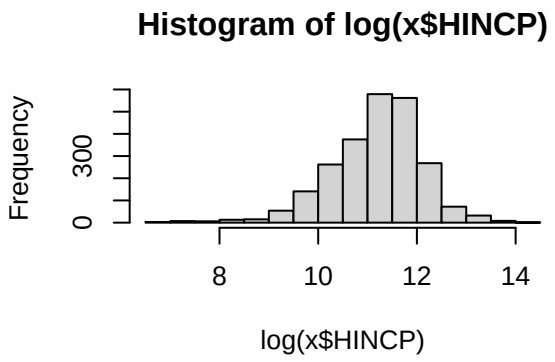


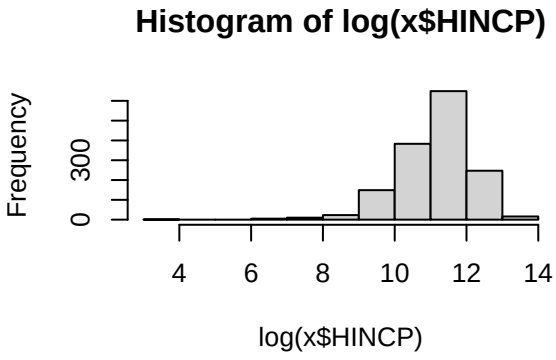


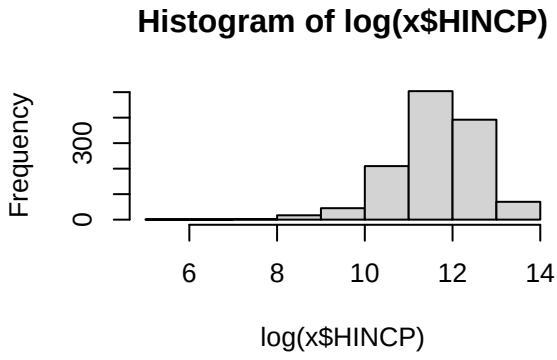



```
## Warning in log(x$HINCP): NaNs produced

## Warning in log(x$HINCP): NaNs produced

## Warning in log(x$HINCP): NaNs produced

## Warning in log(x$HINCP): NaNs produced
```










```
## Warning in log(x$HINCP): NaNs produced
```

	34001	34003	34005	34007
breaks	integer,12	integer,12	integer,14	numeric,19
counts	integer,11	integer,11	integer,13	integer,18
density	numeric,11	numeric,11	numeric,13	numeric,18
mids	numeric,11	numeric,11	numeric,13	numeric,18
xname	"log(x$HINCP)"	"log(x$HINCP)"	"log(x$HINCP)"	"log(x$HINCP)"
equidist	TRUE	TRUE	TRUE	TRUE
	34011	34013	34015	34017
breaks	integer,10	integer,12	integer,11	integer,14
counts	integer,9	integer,11	integer,10	integer,13
density	numeric,9	numeric,11	numeric,10	numeric,13
mids	numeric,9	numeric,11	numeric,10	numeric,13
xname	"log(x$HINCP)"	"log(x$HINCP)"	"log(x$HINCP)"	"log(x$HINCP)"
equidist	TRUE	TRUE	TRUE	TRUE
	34019	34021	34023	34025
breaks	numeric,14	integer,14	integer,13	integer,12
counts	integer,13	integer,13	integer,12	integer,11
density	numeric,13	numeric,13	numeric,12	numeric,11
mids	numeric,13	numeric,13	numeric,12	numeric,11
xname	"log(x$HINCP)"	"log(x$HINCP)"	"log(x$HINCP)"	"log(x$HINCP)"
equidist	TRUE	TRUE	TRUE	TRUE
	34027	34029	34031	34035
breaks	integer,12	numeric,17	integer,12	integer,10
counts	integer,11	integer,16	integer,11	integer,9
density	numeric,11	numeric,16	numeric,11	numeric,9
mids	numeric,11	numeric,16	numeric,11	numeric,9

```
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE           TRUE
##          34037          34039          34041
## breaks   numeric,16     integer,11     integer,11
## counts   integer,15     integer,10     integer,10
## density  numeric,15     numeric,10     numeric,10
## mids     numeric,15     numeric,10     numeric,10
## xname    "log(x$HINCP)" "log(x$HINCP)" "log(x$HINCP)"
## equidist TRUE           TRUE           TRUE
```