DSC520_Week4_Guruprasad_Velikadu_Krishnamoorthy_Part_4.2

Guruprasad Velikadu Krishnamoorthy

2023-01-08

Assignment Week 4 Part 2

Loading the required Packages

```
library(ggplot2)
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(qqplotr)
```

```
##
## Attaching package: 'qqplotr'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     stat_qq_line, StatQqLine
```

```
library(pastecs)
library(readxl)
library(plyr)
library(stringr)
library(lubridate)
```

```
## Loading required package: timechange
```

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

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

Set the working directory to the root of your DSC 520 directory and initial settings

```
knitr::opts_knit$set(root.dir ="C:/Users/Gurup/GURU/Learning/Masters/Term_2/DSC520_T302_Statistics_for_
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 80), tidy = TRUE)
```

4.a.Use the apply function on a variable in your dataset

```
# reading from Excel
excel_path = "data/week-6-housing.xlsx"
housing_df <- read_excel(excel_path)
# Reading only the second column from Dataframe and converting it to a Matrix
salePrice <- as.matrix(housing_df[, 2])
# using the apply function
apply(salePrice, 2, mean)
```

```
## Sale Price
##   660737.7
```

```
apply(salePrice, 2, length)
```

```
## Sale Price
##      12865
```

4.b. Use the aggregate function on a variable in your dataset

```
aggregate(`Sale Price` ~ zip5, housing_df, mean)
```

```
##    zip5 Sale Price
## 1 98052   649375.4
## 2 98053   672623.7
## 3 98059   645000.0
## 4 98074   951543.8
```

```
aggregate(`Sale Price` ~ zip5 + ctyname, housing_df, mean, na.rm = TRUE)
```

```
##    zip5   ctyname Sale Price
## 1 98052   REDMOND   644803.2
## 2 98074 SAMMAMISH   972480.3
```

```
aggregate(cbind(bedrooms, square_feet_total_living) ~ ctyname, housing_df, mean,
    na.rm = TRUE)
```

```
##     ctyname bedrooms square_feet_total_living
## 1   REDMOND 3.683380                 2461.493
## 2 SAMMAMISH 4.090909                 3788.182
```

4.c. Use the plyr function on a variable in your dataset – more specifically, I want to see you split some data, perform a modification to the data, and then bring it back together

Solution: This question is answered with 2 examples, one using ddply where aggregation is done. Other using adply where data manipulation is done for every row.

```
# Using ddply
housingSubset_df1 <- subset(housing_df, grepl("NE", housing_df$addr_full))
AvgsqftLiving <- ddply(housingSubset_df1, .(bedrooms), summarize, meansqft = round(mean(square_feet_tota
AvgSales <- ddply(housingSubset_df1, .(year_built), summarize, meanSales = round(mean(`Sale Price`)))
head(AvgSales)
```

Example 1 : ddply is used to calculate the Average of sales price for houses based on the year built

```
##   year_built meanSales
## 1       1900    394500
## 2       1903    430000
## 3       1905    620000
## 4       1906    550000
## 5       1909      1070
## 6       1910    150000
```

```
# Extracting only 20 sample rows to fit output on screen
housingSubset_df2 <- housingSubset_df1[1:20, ]
# Function to replace the word `NE` with `North East`
replace_NE <- function(x) {
    if (grepl("NE", x)) {
        str1 <- unlist(strsplit(x, " "))
        str2 <- ifelse(str1 == "NE", "North East", str1)
        str2 <- paste(str2, collapse = " ")
        print(str2)
    } else {
        print(x)
    }
}
# Using adply to transform every row in housingSubset_df2
adply(housingSubset_df2$addr_full, 1, function(x) {
    data.frame(updated_addr = replace_NE(x), stringsAsFactors = FALSE)
})
```

Example 2 : adply is used to create a new column that replaces the word "NE" with "North East" in the address column. A function is created to split the data, extract NE out of it , replace it with North East and bring it all together.

```
## [1] "17021 North East 113TH CT"
## [1] "11927 178TH PL North East"
```

```
## [1] "13315 174TH AVE North East"
## [1] "3303 178TH AVE North East"
## [1] "16126 North East 108TH CT"
## [1] "8101 229TH DR North East"
## [1] "21634 North East 87TH PL"
## [1] "21404 North East 67TH ST"
## [1] "7525 238TH AVE North East"
## [1] "17703 North East 26TH ST"
## [1] "14924 North East 74TH CT"
## [1] "7858 148TH CT North East"
## [1] "17905 North East 26TH ST"
## [1] "2921 288TH AVE North East"
## [1] "3624 264TH AVE North East"
## [1] "7850 148TH CT North East"
## [1] "8944 237TH PL North East"
## [1] "11922 173RD PL North East"
## [1] "3201 176TH CT North East"
## [1] "26920 North East 50TH ST"
```

```
##    X1               updated_addr
## 1   1  17021 North East 113TH CT
## 2   2  11927 178TH PL North East
## 3   3 13315 174TH AVE North East
## 4   4  3303 178TH AVE North East
## 5   5  16126 North East 108TH CT
## 6   6   8101 229TH DR North East
## 7   7   21634 North East 87TH PL
## 8   8   21404 North East 67TH ST
## 9   9  7525 238TH AVE North East
## 10 10   17703 North East 26TH ST
## 11 11   14924 North East 74TH CT
## 12 12   7858 148TH CT North East
## 13 13   17905 North East 26TH ST
## 14 14  2921 288TH AVE North East
## 15 15  3624 264TH AVE North East
## 16 16   7850 148TH CT North East
## 17 17   8944 237TH PL North East
## 18 18  11922 173RD PL North East
## 19 19   3201 176TH CT North East
## 20 20   26920 North East 50TH ST
```

4.d.Check distributions of the data

Solution: The distributions are plotted with two examples

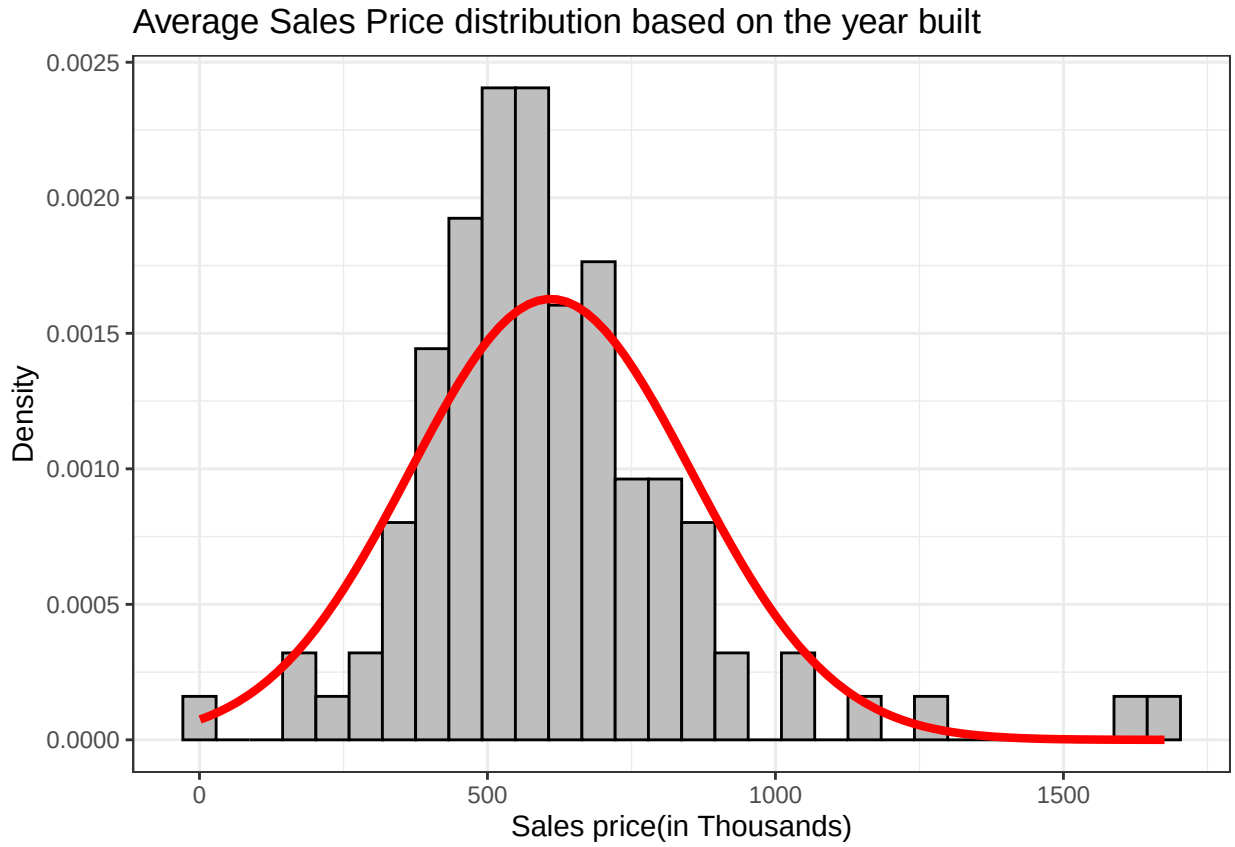
Example 1: Distribution of Average housing sales price based on the year built

```
gg.avgSales <- ggplot(AvgSales, aes(meanSales/1000)) + geom_histogram(aes(y = after_stat(density)),
    color = "black", fill = "grey") + labs(x = "Sales price(in Thousands)", y = "Density",
    title = "Average Sales Price distribution based on the year built")
```

```
gg.avgSales + stat_function(fun = dnorm, args = list(mean = mean(AvgSales$meanSales/1000,
    na.rm = TRUE), sd = sd(AvgSales$meanSales/1000, na.rm = TRUE)), color = "Red",
    linewidth = 1.5) + theme_bw()
```

The Results from Shapiro test suggests the value of p is less than 0.05, hence the distribution is not normal. Also the boxplot has many outliers(shown in next part of the question) which explains the distribution is not normal.The distribution is skewed towards the right. The skewness and kurtosis are also deviated from the values of Normal distribution.

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




```
stat.desc(AvgSales$meanSales, norm = TRUE)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
## 1.080000e+02 0.000000e+00 0.000000e+00 1.070000e+03 1.675500e+06 1.674430e+06
##          sum       median         mean      SE.mean CI.mean.0.95          var
## 6.583972e+07 5.826115e+05 6.096270e+05 2.359771e+04 4.677971e+04 6.014001e+10
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
## 2.452346e+05 4.022699e-01 1.437266e+00 3.090690e+00 4.680859e+00 5.076245e+00
##   normtest.W   normtest.p
## 8.981677e-01 5.129962e-07
```

```r
shapiro.test(AvgSales$meanSales)
```

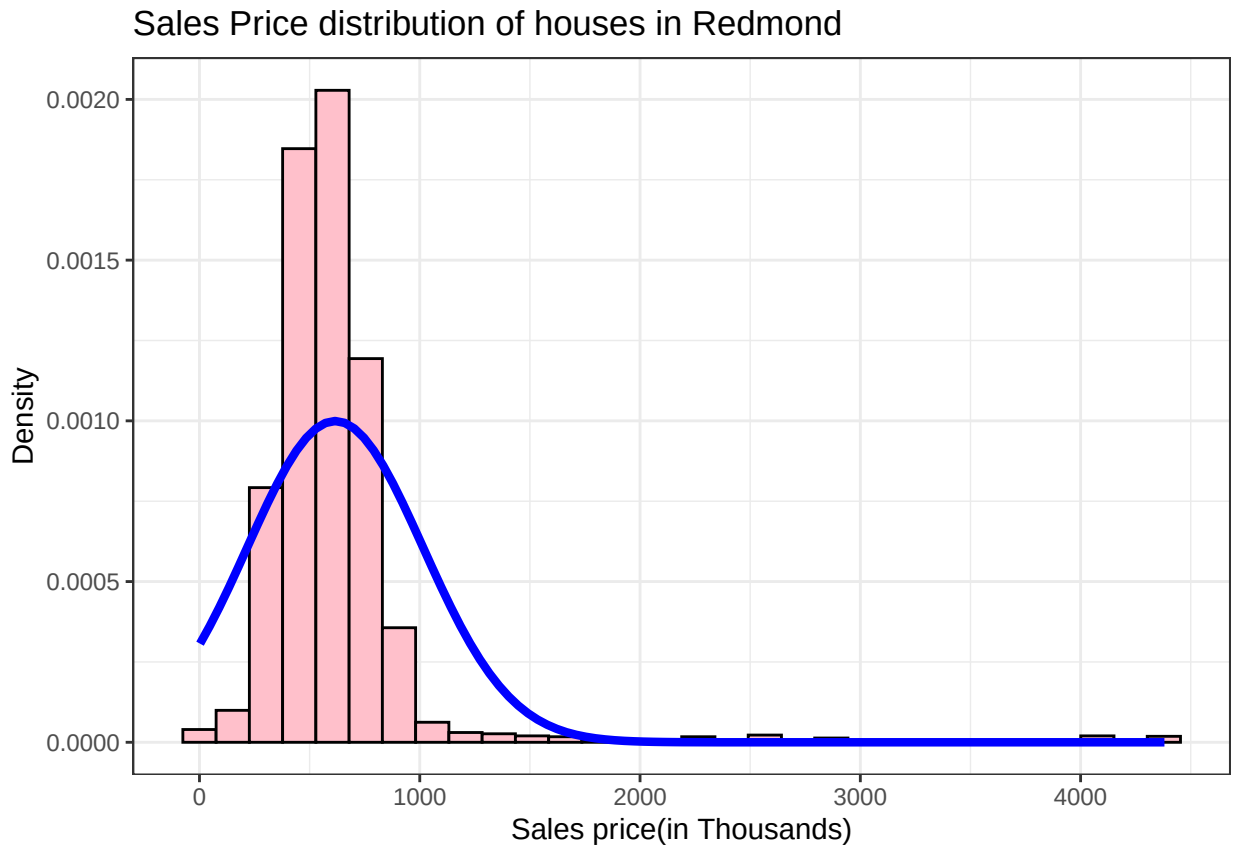
```
##
##  Shapiro-Wilk normality test
##
## data:  AvgSales$meanSales
## W = 0.89817, p-value = 5.13e-07
```

Example 2: Distribution of Housing Sales in Redmond

```r
# Creating a subset of 5000 rows for the houses in Redmond
housingSubset_redmond <- subset(housing_df, housing_df$ctyname == "REDMOND")[1:5000,
    ]
# creating histogram and Normal distribution on the sales price in Redmond
gg.redmond <- ggplot(housingSubset_redmond, aes(`Sale Price`/1000)) + geom_histogram(aes(y = after_stat
    color = "black", fill = "pink") + labs(x = "Sales price(in Thousands)", y = "Density",
    title = "Sales Price distribution of houses in Redmond")
gg.redmond + stat_function(fun = dnorm, args = list(mean = mean(housingSubset_redmond$`Sale Price`/1000
    na.rm = TRUE), sd = sd(housingSubset_redmond$`Sale Price`/1000, na.rm = TRUE)),
    color = "Blue", linewidth = 1.5) + theme_bw()
```

The Results from Shapiro test suggests the value of p is less than 0.05, hence the distribution is not normal. Also the boxplot has many outliers(shown in next part of the question) which explains the distribution is not normal. The distribution is skewed to the right with a long tail

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




```
# Describing the statistics
stat.desc(housingSubset_redmond$`Sale Price`, norm = TRUE)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
## 5.000000e+03 0.000000e+00 0.000000e+00 2.500000e+03 4.380542e+06 4.378042e+06
##          sum       median         mean      SE.mean CI.mean.0.95          var
## 3.079616e+09 5.650000e+05 6.159231e+05 5.644332e+03 1.106537e+04 1.592924e+11
##      std.dev     coef.var     skewness     skew.2SE     kurtosis     kurt.2SE
## 3.991145e+05 6.479941e-01 5.745292e+00 8.295101e+01 4.446853e+01 3.210843e+02
##   normtest.W   normtest.p
## 5.320596e-01 1.008361e-78
```

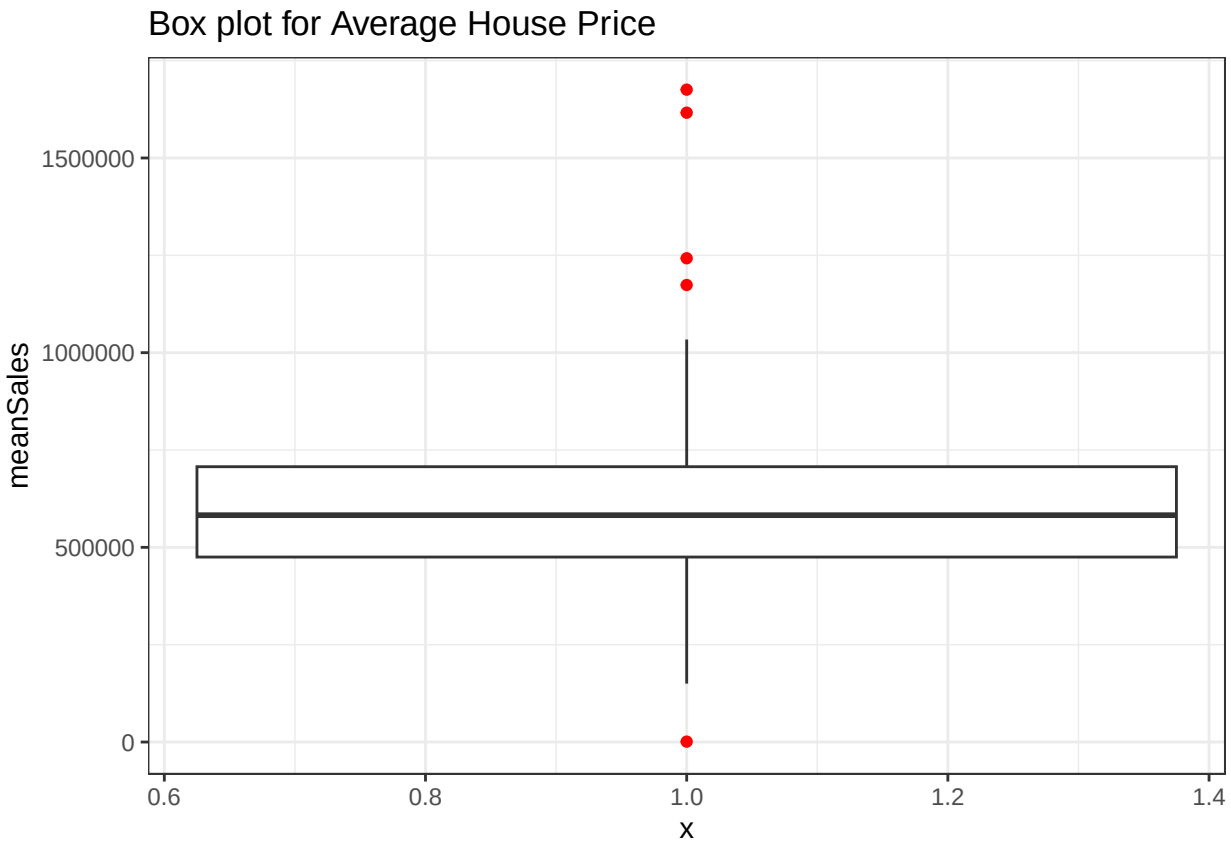
```
shapiro.test(housingSubset_redmond$`Sale Price`)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  housingSubset_redmond$‘Sale Price‘
## W = 0.53206, p-value < 2.2e-16
```

4.e.Identify if there are any outliers

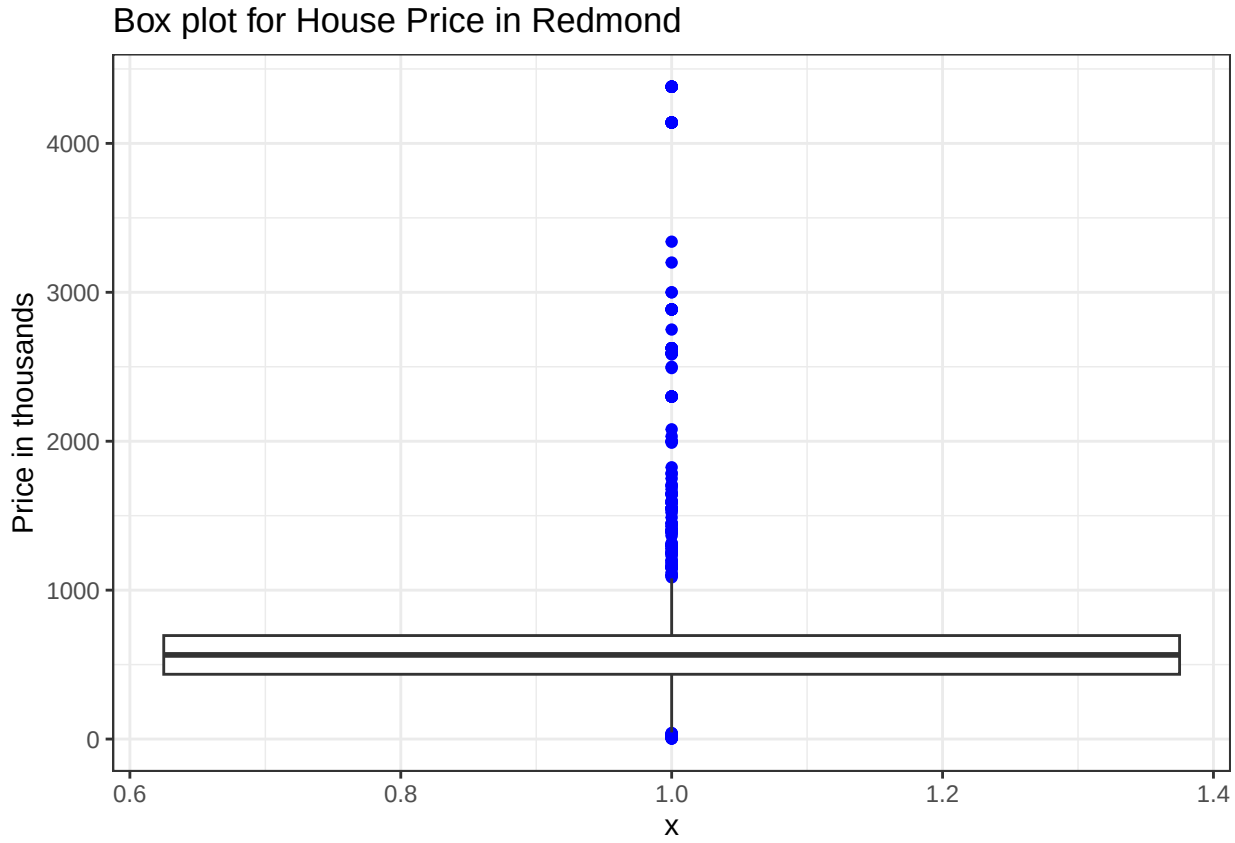
```
ggplot(data = AvgSales, aes(y = meanSales, x = 1)) + geom_boxplot(outlier.colour = "red") +
    theme_bw() + labs(title = "Box plot for Average House Price ")
```

Example 1: The outlers in the boxplot for Average House Price are highlighted in Red




```
ggplot(housingSubset_redmond, aes(y = `Sale Price`/1000, x = 1)) + geom_boxplot(outlier.colour = "blue")
    theme_bw() + labs(title = "Box plot for House Price in Redmond", y = "Price in thousands")
```

Example 2: The outlers in the boxplot for House Price in Redmond are highlighted in Blue



f.Create at least 2 new variables

```
housingSubset_redmond$Sale_Year <- year(housingSubset_redmond$`Sale Date`)
head(housingSubset_redmond$Sale_Year)
```

Variable 1

```
## [1] 2006 2006 2006 2006 2006 2006
```

```
# Using ifelse to create variable
housingSubset_redmond$age_of_prop_when_purchased <- ifelse(housingSubset_redmond$Sale_Year -
    housingSubset_redmond$year_built > 0, housingSubset_redmond$Sale_Year - housingSubset_redmond$year_
    0)
head(housingSubset_redmond$age_of_prop_when_purchased)
```

Variable 2

```
## [1]  3  0 38 26 30 18
```

Session info

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] lubridate_1.9.0  timechange_0.1.1 stringr_1.4.1    plyr_1.8.8
## [5] readxl_1.4.1     pastecs_1.3.21   qqplotr_0.0.5    psych_2.2.9
## [9] ggplot2_3.4.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0  xfun_0.34         lattice_0.20-45   colorspace_2.0-3
##  [5] vctrs_0.5.0       generics_0.1.3    htmltools_0.5.3   yaml_2.3.6
##  [9] utf8_1.2.2        rlang_1.0.6       pillar_1.8.1      glue_1.6.2
## [13] withr_2.5.0       DBI_1.1.3         lifecycle_1.0.3   robustbase_0.95-0
## [17] munsell_0.5.0     gtable_0.3.1      cellranger_1.1.0  evaluate_0.18
## [21] labeling_0.4.2    knitr_1.41        fastmap_1.1.0     parallel_4.2.2
## [25] fansi_1.0.3       highr_0.9         DEoptimR_1.0-11   Rcpp_1.0.9
## [29] scales_1.2.1      formatR_1.12      farver_2.1.1      mnormt_2.1.1
## [33] digest_0.6.30     stringi_1.7.8     dplyr_1.0.10      grid_4.2.2
## [37] cli_3.4.1         tools_4.2.2       magrittr_2.0.3    tibble_3.1.8
## [41] pkgconfig_2.0.3   MASS_7.3-58.1     assertthat_0.2.1  rmarkdown_2.18
## [45] rstudioapi_0.14   R6_2.5.1          boot_1.3-28       nlme_3.1-160
## [49] compiler_4.2.2
```