assignment_4.2_SyversonLuke.R

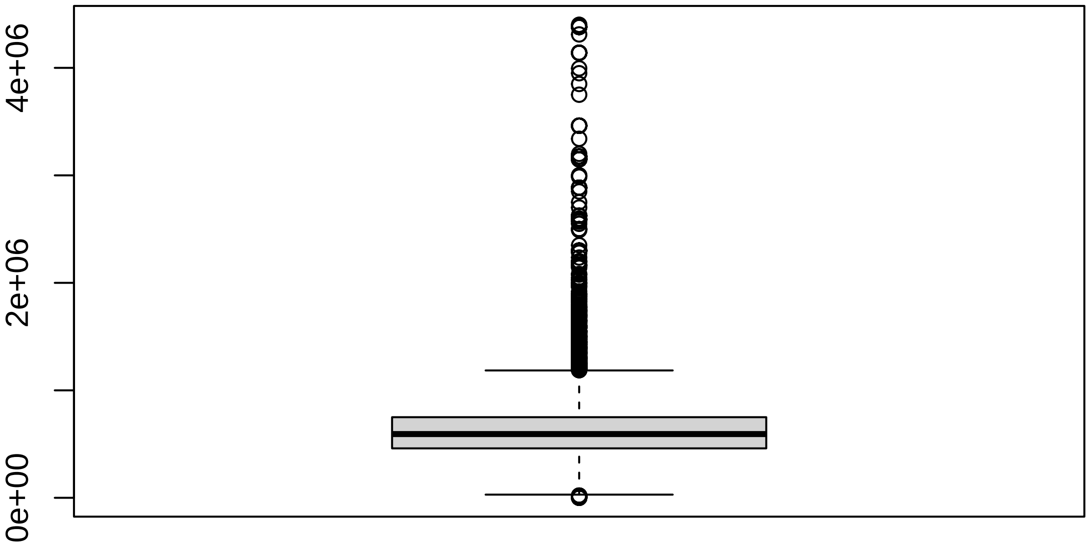
SYVERSONL

2023-04-09

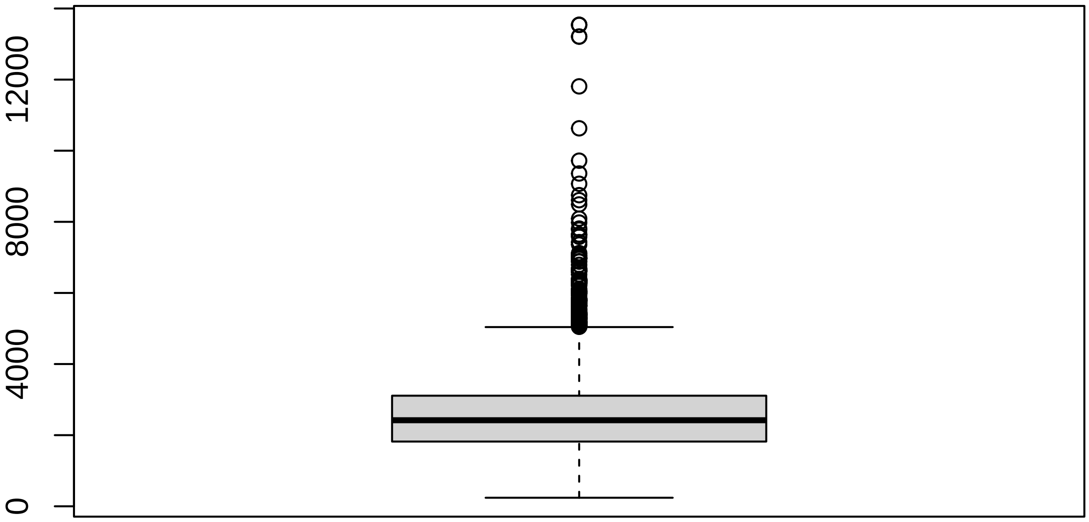
```r
library(readxl)
setwd("~/GitHub/dsc520/assignments/Week4")
t0 <- read_excel("week-6-housing.xlsx")

###
# Use the apply function on a variable in your dataset

ranges <- apply(t0, 2, range)

###
# Use the aggregate function on a variable in your dataset

avg_sale_bedrooms <- aggregate(t0$`Sale Price`, list(t0$bedrooms), mean)

###
# 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

library(plyr)
price_sqft <- function(t0) {fn = t0$`Sale Price`/t0$square_feet_total_living}
#x <- ddply(t0, .variables = 'sale_reason', .fun = price_sqft)

  # The above line is commented so that markdown can run.
  # I can't figure this one out. Oh well. At first I had subsets of the data
  # using the filter(), but rereading the instructions before submission had me
  # attempt to pull this together. The error I get here is:
  # "Results do not have equal length" and I'm not sure why. I'll challenge my
  # understanding of the ddply function this week.

###
# Check distributions of the data

library(pastecs)
stat.desc(t0)
```

```
##                 Sale Date   Sale Price  sale_reason sale_instrument
## nbr.val      1.286500e+04 1.286500e+04 1.286500e+04    1.286500e+04
## nbr.null     0.000000e+00 0.000000e+00 2.000000e+00    3.000000e+00
## nbr.na       0.000000e+00 0.000000e+00 0.000000e+00    0.000000e+00
## min          1.136246e+09 6.980000e+02 0.000000e+00    0.000000e+00
## max          1.481846e+09 4.400000e+06 1.900000e+01    2.700000e+01
## range        3.456000e+08 4.399302e+06 1.900000e+01    2.700000e+01
## sum          1.687715e+13 8.500391e+09 1.994100e+04    4.731400e+04
## median       1.321488e+09 5.930000e+05 1.000000e+00    3.000000e+00
## mean         1.311866e+09 6.607377e+05 1.550019e+00    3.677730e+00
## SE.mean      9.130786e+05 3.565217e+03 2.358588e-02    2.918881e-02
## CI.mean.0.95 1.789770e+06 6.988354e+03 4.623183e-02    5.721441e-02
## var          1.072571e+16 1.635241e+11 7.156721e+00    1.096081e+01
## std.dev      1.035650e+08 4.043811e+05 2.675205e+00    3.310712e+00
## coef.var     7.894483e-02 6.120145e-01 1.725917e+00    9.002051e-01
##              sale_warning sitetype addr_full         zip5 ctyname postalctyn
## nbr.val                NA       NA        NA 1.286500e+04      NA         NA
## nbr.null               NA       NA        NA 0.000000e+00      NA         NA
## nbr.na                 NA       NA        NA 0.000000e+00      NA         NA
## min                    NA       NA        NA 9.805200e+04      NA         NA
## max                    NA       NA        NA 9.807400e+04      NA         NA
## range                  NA       NA        NA 2.200000e+01      NA         NA
## sum                    NA       NA        NA 1.261446e+09      NA         NA
## median                 NA       NA        NA 9.805200e+04      NA         NA
## mean                   NA       NA        NA 9.805254e+04      NA         NA
## SE.mean                NA       NA        NA 1.494488e-02      NA         NA
## CI.mean.0.95           NA       NA        NA 2.929417e-02      NA         NA
## var                    NA       NA        NA 2.873389e+00      NA         NA
## std.dev                NA       NA        NA 1.695107e+00      NA         NA
## coef.var               NA       NA        NA 1.728774e-05      NA         NA
##                        lon          lat building_grade square_feet_total_living
## nbr.val       1.286500e+04 1.286500e+04   1.286500e+04             1.286500e+04
## nbr.null      0.000000e+00 0.000000e+00   0.000000e+00             0.000000e+00
## nbr.na        0.000000e+00 0.000000e+00   0.000000e+00             0.000000e+00
## min          -1.221643e+02 4.745635e+01   2.000000e+00             2.400000e+02
## max          -1.219499e+02 4.773255e+01   1.300000e+01             1.354000e+04
## range         2.144216e-01 2.761993e-01   1.100000e+01             1.330000e+04
## sum          -1.570549e+06 6.134492e+05   1.060130e+05             3.267075e+07
## median       -1.221003e+02 4.768742e+01   8.000000e+00             2.420000e+03
## mean         -1.220792e+02 4.768358e+01   8.240420e+00             2.539506e+03
## SE.mean       4.603069e-04 2.271998e-04   9.633091e-03             8.726704e+00
## CI.mean.0.95  9.022698e-04 4.453453e-04   1.888229e-02             1.710564e+01
## var           2.725867e-03 6.640879e-04   1.193826e+00             9.797388e+05
## std.dev       5.220984e-02 2.576990e-02   1.092624e+00             9.898176e+02
## coef.var     -4.276718e-04 5.404356e-04   1.325932e-01             3.897677e-01
##                  bedrooms bath_full_count bath_half_count bath_3qtr_count
## nbr.val      1.286500e+04    1.286500e+04    1.286500e+04    1.286500e+04
## nbr.null     1.900000e+01    5.100000e+01    5.177000e+03    7.457000e+03
## nbr.na       0.000000e+00    0.000000e+00    0.000000e+00    0.000000e+00
## min          0.000000e+00    0.000000e+00    0.000000e+00    0.000000e+00
## max          1.100000e+01    2.300000e+01    8.000000e+00    8.000000e+00
## range        1.100000e+01    2.300000e+01    8.000000e+00    8.000000e+00
## sum          4.475300e+04    2.313700e+04    7.891000e+03    6.355000e+03
## median       4.000000e+00    2.000000e+00    1.000000e+00    0.000000e+00
## mean         3.478663e+00    1.798445e+00    6.133696e-01    4.939759e-01
## SE.mean      7.724356e-03    5.737733e-03    4.639903e-03    5.731102e-03
## CI.mean.0.95 1.514088e-02    1.124681e-02    9.094899e-03    1.123381e-02
## var          7.675990e-01    4.235361e-01    2.769668e-01    4.225578e-01
## std.dev      8.761273e-01    6.507965e-01    5.262763e-01    6.500444e-01
## coef.var     2.518575e-01    3.618662e-01    8.580085e-01    1.315944e+00
##                year_built year_renovated current_zoning    sq_ft_lot prop_type
## nbr.val      1.286500e+04   1.286500e+04             NA 1.286500e+04        NA
## nbr.null     0.000000e+00   1.269600e+04             NA 0.000000e+00        NA
## nbr.na       0.000000e+00   0.000000e+00             NA 0.000000e+00        NA
## min          1.900000e+03   0.000000e+00             NA 7.850000e+02        NA
## max          2.016000e+03   2.016000e+03             NA 1.631322e+06        NA
## range        1.160000e+02   2.016000e+03             NA 1.630537e+06        NA
## sum          2.563998e+07   3.376330e+05             NA 2.859705e+08        NA
## median       1.998000e+03   0.000000e+00             NA 7.965000e+03        NA
## mean         1.993003e+03   2.624431e+01             NA 2.222857e+04        NA
## SE.mean      1.518212e-01   2.005595e+00             NA 5.019511e+02        NA
## CI.mean.0.95 2.975921e-01   3.931264e+00             NA 9.838986e+02        NA
## var          2.965342e+02   5.174832e+04             NA 3.241400e+09        NA
## std.dev      1.722017e+01   2.274826e+02             NA 5.693329e+04        NA
## coef.var     8.640314e-03   8.667883e+00             NA 2.561267e+00        NA
##               present_use
## nbr.val      1.286500e+04
## nbr.null     9.000000e+00
## nbr.na       0.000000e+00
## min          0.000000e+00
## max          3.000000e+02
## range        3.000000e+02
## sum          8.488000e+04
## median       2.000000e+00
## mean         6.597746e+00
## SE.mean      2.663628e-01
## CI.mean.0.95 5.221105e-01
## var          9.127604e+02
## std.dev      3.021192e+01
## coef.var     4.579128e+00
```

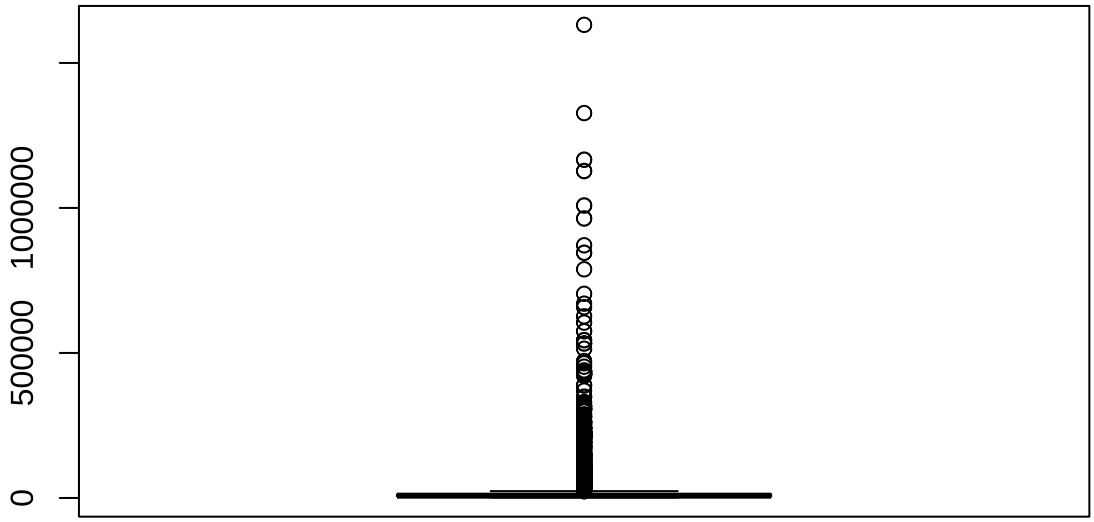
```
boxplot(t0$`Sale Price`)
```



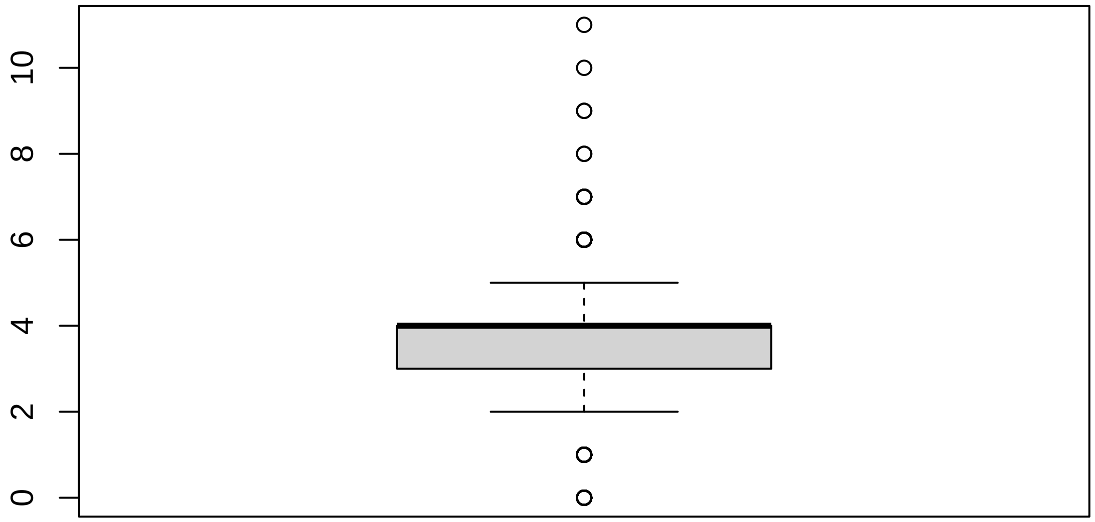
```
boxplot(t0$square_feet_total_living)
```



```
boxplot(t0$sq_ft_lot)
```



```
boxplot(t0$bedrooms)
```



```
###
# Identify if there are any outliers

  # It looks like the data has outliers due to its general quantitative right-skewness.
  # It would make sense that the majority of houses sold for less, had less
  # amenities and smaller sizes.

###
# Create at least 2 new variables

detach('package:plyr')
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:pastecs':
##
##     first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
```

```
##
##     intersect, setdiff, setequal, union
```

```
mean_price <- mean(t0$`Sale Price`)
t.sammamish <- distinct(filter(t0, ctyname == 'SAMMAMISH'))
mean_price_sammamish <- mean(t.sammamish$`Sale Price`)
```