5: Bootstrap Method Revisited and Bootstrap MLR

Cheng Peng

West Chester University

Contents

1 Bootstrap Method Revisited

The **bootstrap** is one of the computationally intensive techniques that is now part of the broad umbrella of **nonparametric statistics** that are commonly called **resampling methods**.

We have used this method to construct the confidence interval for a unknown population mean and correlation coefficient of two numerical variables (populations). We also fit bootstrap simple linear regression in two different ways and will use it to build bootstrap MLR.

Based on the limited experience using the bootstrap procedure we had, it is time to look into the resampling technique a little more closer to how and why it works. We will also discuss the implicit assumptions for bootstrap resampling procedures. For ease of illustration, we need to review some concepts learned earlier: probability density function (`pdf`) and cumulative distribution function (`CDF`).

1.1 `pdf` vs `CDF`

Let $f(x)$ denote the density function of the distribution of the population of interest. The corresponding cumulative distribution is given by

$$F(x) = \int_{-\infty}^{x} f(y)dy$$

That is, $F'(x) = f(x)$. The following animated graph illustrates the relationship geometrically.

https://github.com/pengdsci/STA504/blob/main/topic03/w04-PDFvsCDF.gif?raw=true

1.2 Comparing Two Distributions

Next, we use two examples to show how to do visual comparison between distributions.

```r
par(mfrow=c(1,2))
x = seq(0, 20, length = 200)
plot(x, dnorm(x,5,1), type="l", xlab = "", ylab = "",
        main = "Comparing Two Normal Density Curves",
        cex.main = 0.8, col.main = "navy", col = "red", lty=1, lwd = 2)
lines(x, dnorm(x, 7, 2), col = "blue", lty =2, lwd = 2)
legend("right", c("N(5,1)", "N(7,2)"), col = c("red", "blue"), lty=1:2,
                    lwd = rep(2,2), cex = 0.8, bty="n")
###
dist = abs(pnorm(x, 7, 2) - pnorm(x,5,1))
max.id = which(dist == max(dist))
plot(x, pnorm(x,5,1), type="l", xlab = "", ylab = "",
        main = "Comparing Two Normal CDF Curves",
        cex.main = 0.8, col.main = "navy", col = "red", lty=1, lwd = 2)
lines(x, pnorm(x, 7, 2), col = "blue", lty=2, lwd = 2)
segments(x[63],pnorm(x, 7, 2)[63], x[63], pnorm(x,5,1)[63], col = "purple", lwd = 2)
points(rep(x[63],2), c(pnorm(x, 7, 2)[63], pnorm(x,5,1)[63]), pch=19, cex = 1.5, col = "purple")
###
legend("right", c("N(5,1)", "N(7,2)"), col = c("red", "blue"), lty=1:2,
                    lwd = rep(2,2), cex = 0.8, bty="n")
```

We can see from the above figure that the **maximum distance between the two CDF curves** (the length of the purple line segment) measures that discrepancy the two distributions.

1.3 Empirical Distribution and CDF

An empirical distribution is a non-parametric estimation of a theoretical distribution based on a random sample taken from the population. Assume that $\{x_1, x_2, x_3, \cdots, x_n\}$ is a random sample taken from a population with CDF $F(x|\theta)$ (θ is the population parameter). **The empirical distribution of $F(x|\theta)$ based on the given random sample is defined to be

$$F_n(x) = \frac{\text{number of values in random sample } \le x}{n}$$

For example, let's look at the emprical distribution based on the toy data set $\{1, 1.4, 2.1, 2.5, 5\}$.

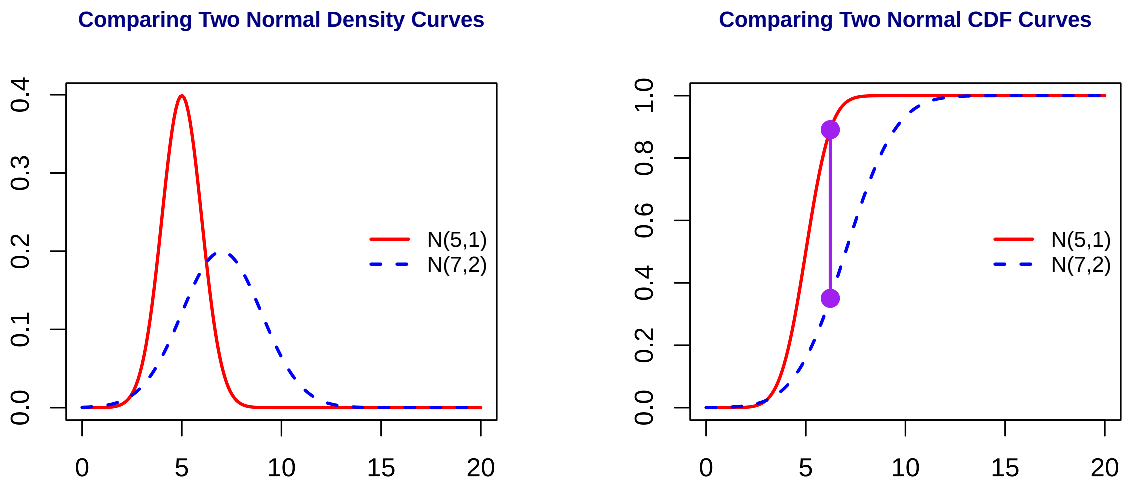
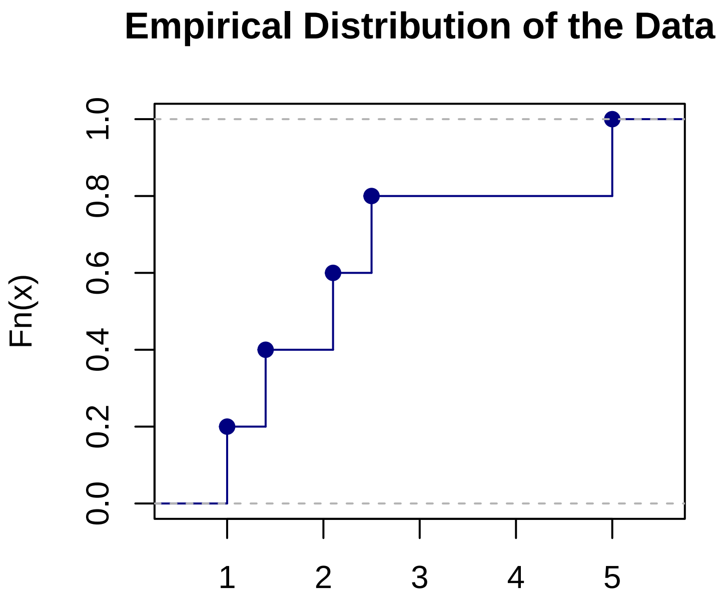
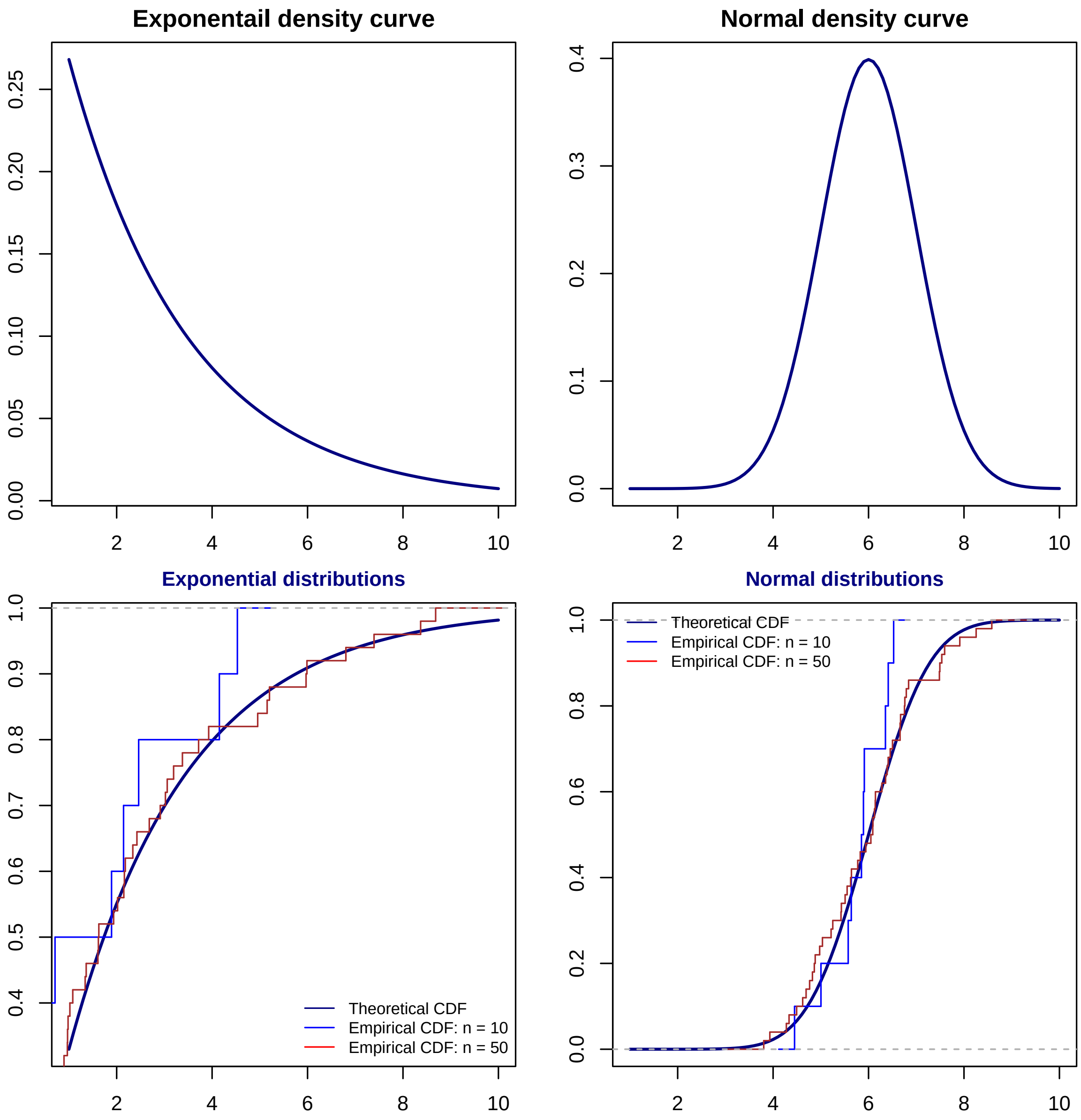


Figure 1: Comparison between two normal distributions



1.4 Relationship between $F(x)$ and $F_n(X)$

As stated earlier, An empirical distribution, $F_n(x)$, is used to estimate the theoretical CDF, $F(x)$. Next, we use several examples to visually check the goodness of the estimation.



We can see two general but important facts:

1. The goodness of the estimation is dependent on the sample size. The bigger the sample size, the more accurate the estimation.

2. With the same size, the goodness of estimation is dependent on the population distribution.

1.5 Why Bootstrap Procedure Works?

For a given population, if the sample size is large enough so that the empirical distribution is close to the theoretical distribution, we can take bootstrap samples for non-parametric inference. The following chart explains how bootstrap method works.

The above

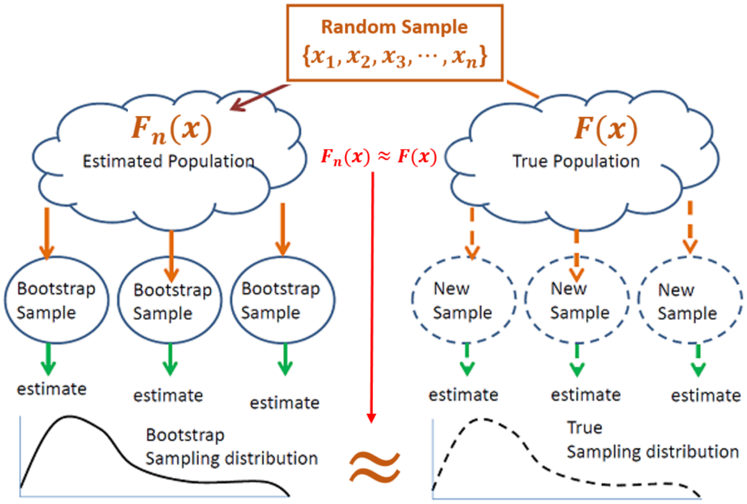


Figure 2: How bootstrap procedure works.

1.6 A Bigger Picture

2 Other Resampling Methods

In the last module, we listed several goodness-of-fit measures. The **predicted residual error sum of squares (PRESS)** is one of them. It definition is not clearly given. Before we discuss this measure in little bit more details, we first introduce a popular resampling method - Jackknife.

2.1 Jackknif Resampling

The jackknife estimator of a parameter is found by systematically **leaving out each observation from a dataset** and **calculating the parameter estimate over the remaining observations** and **then aggregating these calculations**. It was developed by Maurice Quenouille in 1949 and refined in 1956. John Tukey expanded on the technique in 1958 and name the method "jackknife" because it is handy like a physical jack-knife. Jackknife resampling can improvise a solution for many specific problems more efficiently than some purpose-designed tools.

The basic leave-one-out Jackknife resampling method is illustrated in the following chart.

2.2 Jackknife for Predictive Sum of Squares (PRESS)

A fitted model having been produced, each observation in turn is removed and the model is refitted using the remaining observations. The out-of-sample predicted value is calculated for the omitted observation in each case, and the PRESS statistic is calculated as the sum of the squares of all the resulting prediction errors.

$$\text{PRESS} = \sum_{i=1}^{n} (y_i - \hat{y}_{i,-i})^2$$

The PRESS statistic can be calculated for a number of candidate models for the same dataset, with the lowest values of PRESS indicating the best model.

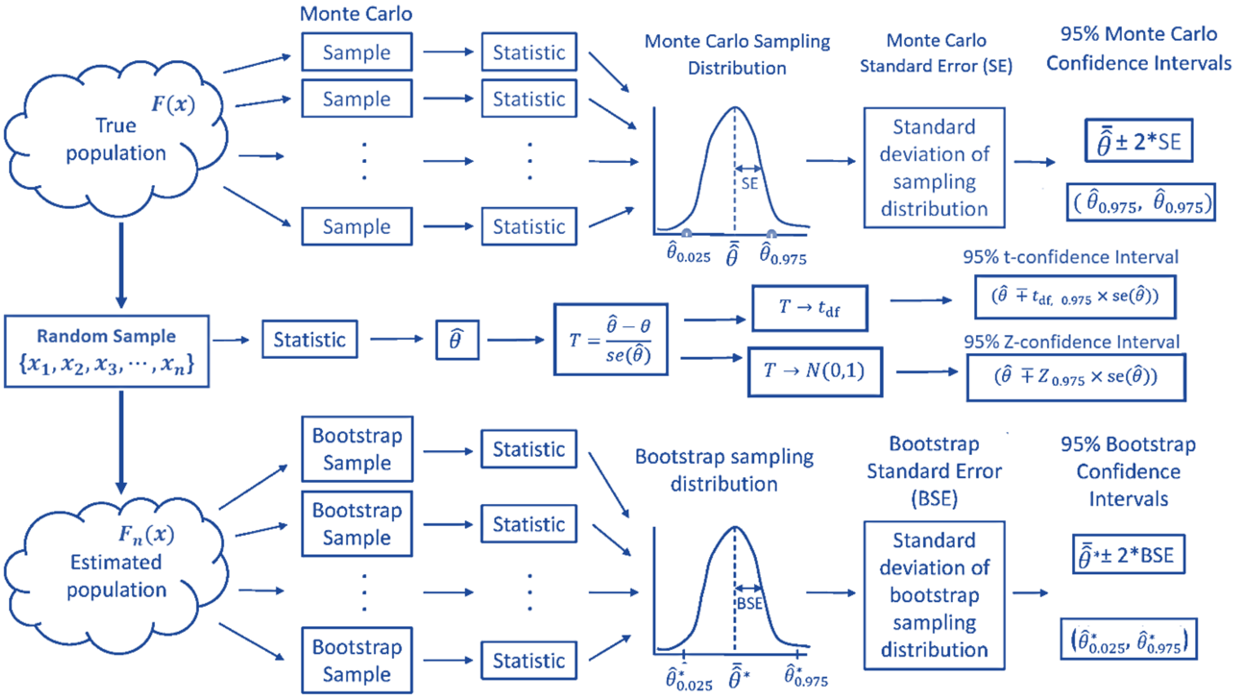

Monte Carlo
Sample
Sample
Sample
Statistic
Statistic
Statistic
$F(x)$
True population
Monte Carlo Sampling Distribution
SE
$\hat{\theta}_{0.025}$ $\bar{\hat{\theta}}$ $\hat{\theta}_{0.975}$
Monte Carlo Standard Error (SE)
Standard deviation of sampling distribution
95% Monte Carlo Confidence Intervals
$\bar{\hat{\theta}} \pm 2*\text{SE}$
$(\hat{\theta}_{0.975}, \hat{\theta}_{0.975})$
Random Sample $\{x_1, x_2, x_3, \cdots, x_n\}$
Statistic
$\hat{\theta}$
$T = \frac{\hat{\theta} - \theta}{se(\hat{\theta})}$
$T \to t_{df}$
$T \to N(0,1)$
95% t-confidence Interval
$(\hat{\theta} \mp t_{df,\ 0.975} \times se(\hat{\theta}))$
95% Z-confidence Interval
$(\hat{\theta} \mp Z_{0.975} \times se(\hat{\theta}))$
$F_n(x)$
Estimated population
Bootstrap Sample
Bootstrap Sample
Bootstrap Sample
Statistic
Statistic
Statistic
Bootstrap sampling distribution
BSE
$\hat{\theta}^*_{0.025}$ $\bar{\hat{\theta}}^*$ $\hat{\theta}^*_{0.975}$
Bootstrap Standard Error (BSE)
Standard deviation of bootstrap sampling distribution
95% Bootstrap Confidence Intervals
$\bar{\hat{\theta}}^* \pm 2*\text{BSE}$
$(\hat{\theta}^*_{0.025}, \hat{\theta}^*_{0.975})$

Figure 3: Monte Carlo, Bootstrap, parametric, and asymptoic confidence intervals

```r
realestate <- read.csv("https://raw.githubusercontent.com/pengdsci/sta321/main/ww03/w03-Realestate.csv"
unitPrice <- realestate$PriceUnitArea
HouseAge <- realestate$HouseAge
nn <- length(HouseAge)
JK.pred <- NULL
for(i in 1:nn){
  unitPrice0 <- unitPrice[-i]
  HouseAge0 <- HouseAge[-i]
  mm <- lm(unitPrice0 ~ HouseAge0)
  JK.pred[i] = as.vector(predict(mm, newdata = data.frame(HouseAge0 = HouseAge[i])))
}
PRESS = sum((unitPrice-JK.pred)^2)
PRESS
```

```
## [1] 73815.25
```

2.3 Some Comments on Jackknife

- Jackknife samples can also be used for estimating population parameters just like what we did in bootstrap methods, but the procedure is not as straightforward as that of bootstrap. We will not go to details in this direction.
- The logic of Jackknife sampling is used to define various data-driven methods that used in modern statistics, data science and machine learning. One of such methods is **cross-validation**. We will introduce this method in the second part this course - generalized linear regression.

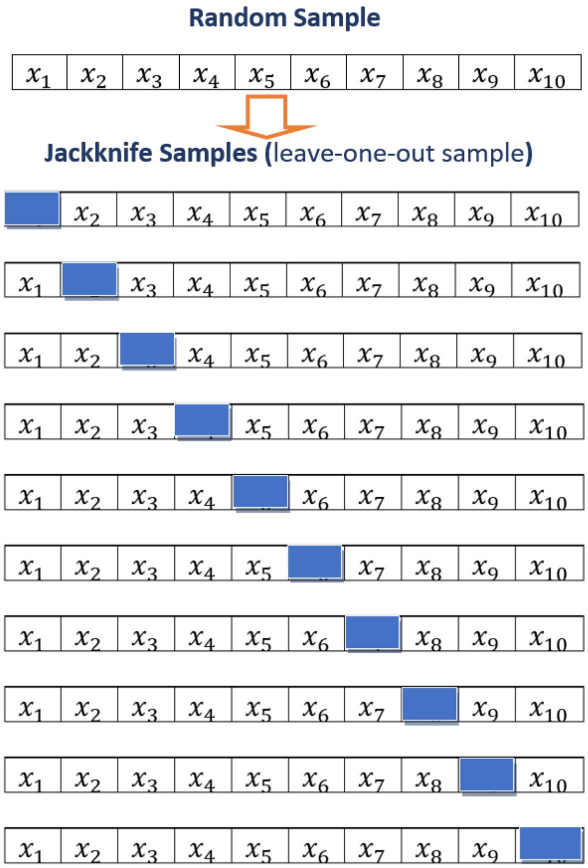

Random Sample
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10}
Jackknife Samples (leave-one-out sample)
x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10}
x_1 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10}
x_1 x_2 x_4 x_5 x_6 x_7 x_8 x_9 x_{10}
x_1 x_2 x_3 x_5 x_6 x_7 x_8 x_9 x_{10}
x_1 x_2 x_3 x_4 x_6 x_7 x_8 x_9 x_{10}
x_1 x_2 x_3 x_4 x_5 x_7 x_8 x_9 x_{10}
x_1 x_2 x_3 x_4 x_5 x_6 x_8 x_9 x_{10}
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_9 x_{10}
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_{10}
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9

Figure 4: Illustration of leave-one-out Jackknife resampling method

3 Case Study - Factors That Affect House Prices

In this note, we introduce two versions of bootstrap procedures to generate bootstrap samples to estimate the confidence intervals of the coefficients of the regression model identified last week. For reference, The code that creates the analytic data set for the final model and the summarized statistics of the model was included in the note.

3.1 Models with Transformed Response

As mentioned earlier, the price is usually skewed to the right. We will simply tale logarithmic transformation of the price. We could also try Box-cox transformation to explore other potential transformations.

```
realestate0 <- read.csv("https://raw.githubusercontent.com/pengdsci/sta321/main/ww03/w03-Realestate.csv
realestate <- realestate0[, -1]
# longitude and latitude will be used to make a map in the upcoming analysis.
lat <- realestate$Latitude
lon <- realestate$Longitude
##
geo.group <- (lon > 121.529) & (lat > 24.96)     # define the geo.group variable
                                                 # top-right region = TRUE, other region = FALSE
realestate$geo.group <- as.character(geo.group)  # convert the logical values to character values.
realestate$sale.year <- as.character(realestate$TransactionYear) # convert transaction year to dummy.
realestate$Dist2MRT.kilo <- (realestate$Distance2MRT)/1000   # re-scale distance: foot -> kilo feet
final.data = realestate[, -c(1,3,5,6)]          # keep only variables to be used in the candidate model
final.data$logAreaPrice = log(final.data$PriceUnitArea)  #
## the final model
log.price <- lm(log(PriceUnitArea) ~ HouseAge + NumConvenStores + sale.year +
                 Dist2MRT.kilo  + geo.group, data = final.data)
log.price02 <- lm(logAreaPrice ~ HouseAge + NumConvenStores + sale.year +
                 Dist2MRT.kilo  + geo.group, data = final.data)
cmtrx <- summary(log.price)$coef
cmtrx02 <- summary(log.price02)$coef
kable(cmtrx, caption = "Inferential Statistics of Final Model")
```

Table 1: Inferential Statistics of Final Model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.5724282	0.0443235	80.599030	0.0000000
HouseAge	-0.0075712	0.0010097	-7.498507	0.0000000
NumConvenStores	0.0274872	0.0049096	5.598667	0.0000000
sale.year2013	0.0805519	0.0244373	3.296272	0.0010655
Dist2MRT.kilo	-0.1445122	0.0137541	-10.506820	0.0000000
geo.groupTRUE	0.1825871	0.0347151	5.259583	0.0000002

The explicit expression of the final model is given by

$$\log(\text{price}) = 3.5723 - 0.0076 \times \text{HouseAge} + 0.0275 \times \text{NumConvenStores}+$$

$$0.0805 \times \text{Sale.year2013} - 0.1445 \times \text{Dist2MRT.kilo} + 0.1826 \times \text{geo.groupTRUE}$$

To interpret the regression coefficient, we choose the coefficient associated with **geo.group**. In the output, you see the name of the dummy variable with the suffix `TRUE`, **geo.groupTRUE**. The suffix **TRUE** indicates that the dummy variable represents the category 'TRUE' of the category variable **geo.group**. The associated coefficient reflects the **mean** difference between the category `TRUE` and the baseline category `FALSE`. In R, the default baseline category is the lowest value of the categorical variable (in alphabetical order).

Let's consider **the set of all houses** that are in the same conditions except the regions (region `TRUE` and region `FALSE`) and the sale prices.

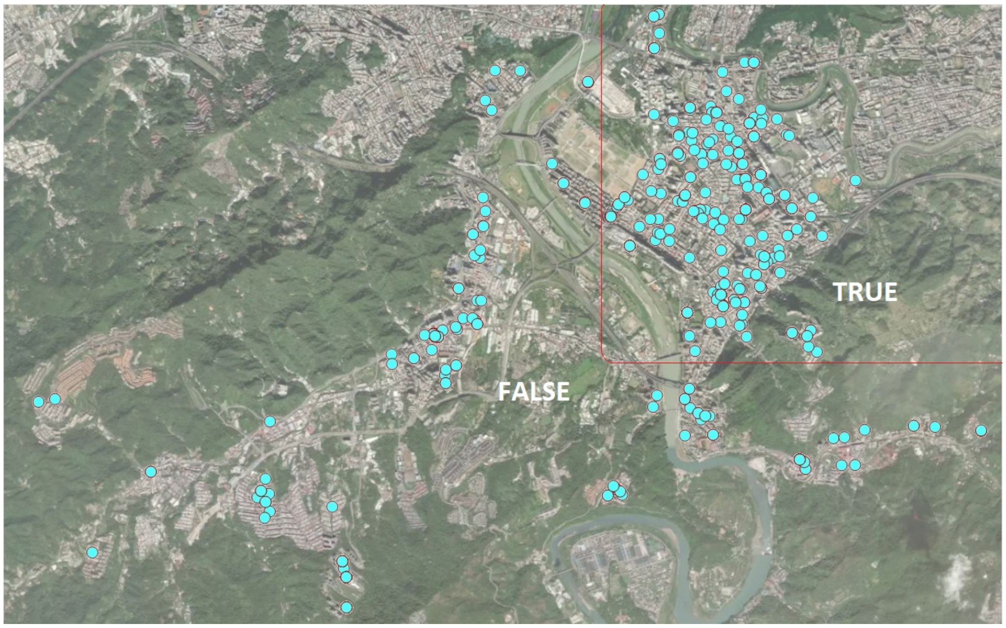


Figure 5: Location of Houses for Sale

3.2 Bootstrap Cases (BOOT.C)

In this section, we use bootstrapping cases to find the confidence intervals for the coefficients in the final regression model. The method was used in bootstrap simple linear regression (SLR) in week #3. The following code finds the confidence interval.

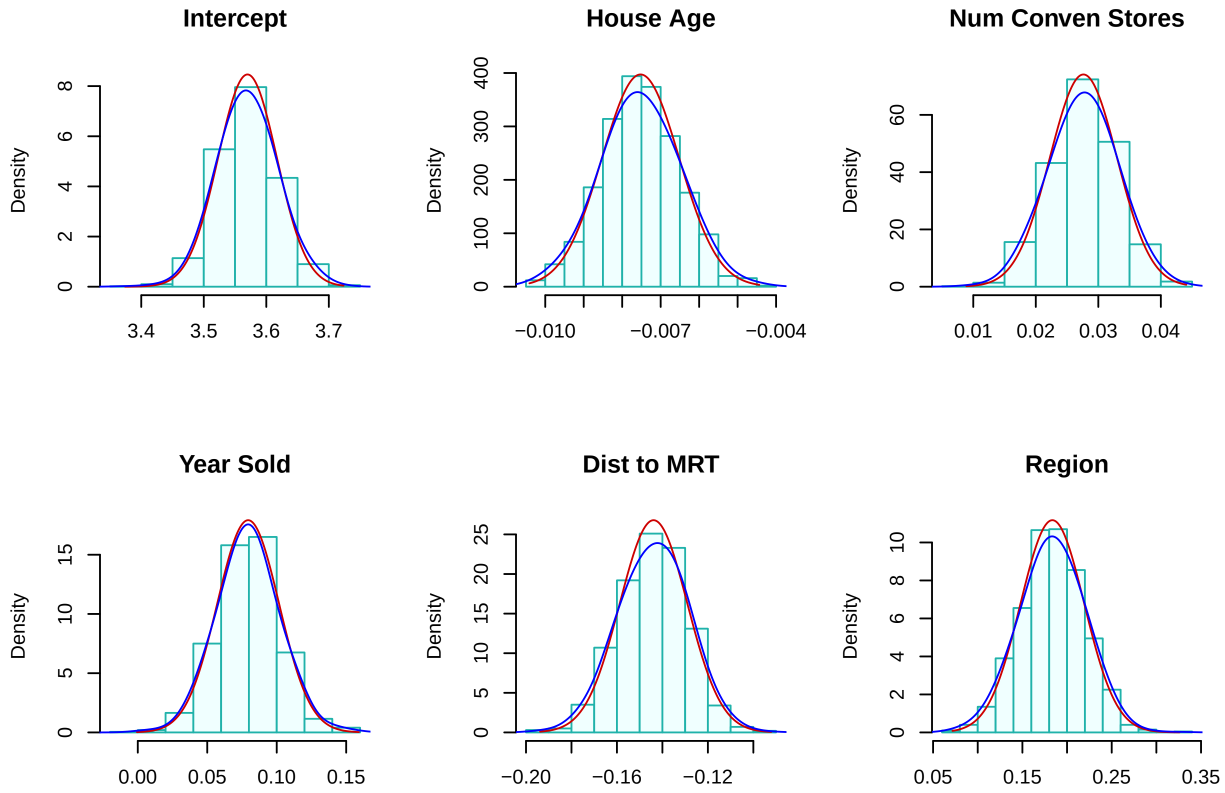
```
log.price = lm(log(PriceUnitArea) ~ HouseAge + NumConvenStores + sale.year +
                 Dist2MRT.kilo  + geo.group, data = final.data)
##
B = 1000       # choose the number of bootstrap replicates.
##
num.p = dim(model.frame(log.price))[2]  # returns number of parameters in the model
smpl.n = dim(model.frame(log.price))[1] # sample size
## zero matrix to store bootstrap coefficients
coef.mtrx = matrix(rep(0, B*num.p), ncol = num.p)
##
for (i in 1:B){
  bootc.id = sample(1:smpl.n, smpl.n, replace = TRUE) # fit final model to the bootstrap sample
  log.price.btc = lm(log(PriceUnitArea) ~ HouseAge + NumConvenStores + sale.year +
                       Dist2MRT.kilo  + geo.group, data = final.data[bootc.id,])
  coef.mtrx[i,] = coef(log.price.btc)    # extract coefs from bootstrap regression model
}
```

We define an R function to make histograms of the bootstrap regression coefficients in the following. I will also use this function to make histograms for the residual bootstrap estimated regression coefficients as well.

```
boot.hist = function(cmtrx, bt.coef.mtrx, var.id, var.nm){
  ## bt.coef.mtrx = matrix for storing bootstrap estimates of coefficients
  ## var.id = variable ID (1, 2, ..., k+1)
  ## var.nm = variable name on the hist title, must be the string in the double quotes
  ## coefficient matrix of the final model
  ## Bootstrap sampling distribution of the estimated coefficients
  x1.1 <- seq(min(bt.coef.mtrx[,var.id]), max(bt.coef.mtrx[,var.id]), length=300 )
  y1.1 <- dnorm(x1.1, mean(bt.coef.mtrx[,var.id]), sd(bt.coef.mtrx[,var.id]))
  # height of the histogram - use it to make a nice-looking histogram.
  highestbar = max(hist(bt.coef.mtrx[,var.id], plot = FALSE)$density)
  ylimit <- max(c(y1.1,highestbar))
  hist(bt.coef.mtrx[,var.id], probability = TRUE, main = var.nm, xlab="",
       col = "azure1",ylim=c(0,ylimit), border="lightseagreen")
  lines(x = x1.1, y = y1.1, col = "red3")
  lines(density(bt.coef.mtrx[,var.id], adjust=2), col="blue")
  #legend("topright", c(""))
}
```

The following histograms of the bootstrap estimates of regression coefficients represent the sampling distributions of the corresponding estimates in the final model.

```
par(mfrow=c(2,3))  # histograms of bootstrap coefs
boot.hist(bt.coef.mtrx=coef.mtrx, var.id=1, var.nm ="Intercept" )
boot.hist(bt.coef.mtrx=coef.mtrx, var.id=2, var.nm ="House Age" )
boot.hist(bt.coef.mtrx=coef.mtrx, var.id=3, var.nm ="Num Conven Stores" )
boot.hist(bt.coef.mtrx=coef.mtrx, var.id=4, var.nm ="Year Sold" )
boot.hist(bt.coef.mtrx=coef.mtrx, var.id=5, var.nm ="Dist to MRT" )
boot.hist(bt.coef.mtrx=coef.mtrx, var.id=6, var.nm ="Region" )
```



Two normal-density curves were placed on each of the histograms.

- The red **density curve** uses the estimated regression coefficients and their corresponding standard error in the output of the regression procedure. The p-values reported in the output are based on the red curve.
- The **blue curve** is a non-parametric data-driven estimate of the density of bootstrap sampling distribution. The bootstrap confidence intervals of the regressions are based on these non-parametric bootstrap sampling distributions.

We can see from the above histograms that the two density curves in all histograms are close to each other. we would expect that significance test results and the corresponding bootstrap confidence intervals are consistent. Next, we find 95% bootstrap confidence intervals of each regression coefficient and combined them with the output of the final model.

```
num.p = dim(coef.mtrx)[2]  # number of parameters
btc.ci = NULL
btc.wd = NULL
for (i in 1:num.p){
  lci.025 = round(quantile(coef.mtrx[, i], 0.025, type = 2),8)
  uci.975 = round(quantile(coef.mtrx[, i],0.975, type = 2 ),8)
  btc.wd[i] =  uci.975 - lci.025
  btc.ci[i] = paste("[", round(lci.025,4),", ", round(uci.975,4),"]")
 }
#as.data.frame(btc.ci)
kable(as.data.frame(cbind(formatC(cmtrx,4,format="f"), btc.ci.95=btc.ci)),
      caption = "Regression Coefficient Matrix")
```

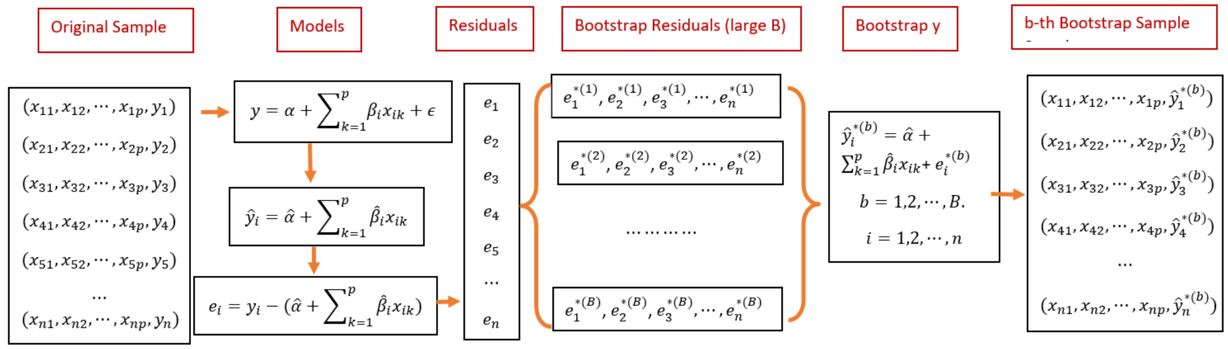
Table 2: Regression Coefficient Matrix

	Estimate	Std. Error	t value	Pr(>|t|)	btc.ci.95
(Intercept)	3.5724	0.0443	80.5990	0.0000	[3.4833 , 3.6671]
HouseAge	-0.0076	0.0010	-7.4985	0.0000	[-0.0096 , -0.0056]
NumConvenStores	0.0275	0.0049	5.5987	0.0000	[0.017 , 0.0379]
sale.year2013	0.0806	0.0244	3.2963	0.0011	[0.0365 , 0.1218]
Dist2MRT.kilo	-0.1445	0.0138	-10.5068	0.0000	[-0.1744 , -0.1149]
geo.groupTRUE	0.1826	0.0347	5.2596	0.0000	[0.1133 , 0.2507]

We can see from the above table of summarized statistics, the significance tests of regression coefficients based on the p-values and the corresponding 95% confidence intervals are consistent.

3.3 Bootstrap Residuals (BOOT.R)

In this section, we introduce bootstrap residual methods to construct bootstrap confidence intervals. The idea is straightforward and is summarized in the following.


Original Sample
Models
Residuals
Bootstrap Residuals (large B)
Bootstrap y
b-th Bootstrap Sample
$(x_{11}, x_{12}, \cdots, x_{1p}, y_1)$
$(x_{21}, x_{22}, \cdots, x_{2p}, y_2)$
$(x_{31}, x_{32}, \cdots, x_{3p}, y_3)$
$(x_{41}, x_{42}, \cdots, x_{4p}, y_4)$
$(x_{51}, x_{52}, \cdots, x_{5p}, y_5)$
$\cdots$
$(x_{n1}, x_{n2}, \cdots, x_{np}, y_n)$
$y = \alpha + \sum_{k=1}^{p} \beta_i x_{ik} + \epsilon$
$\hat{y}_i = \hat{\alpha} + \sum_{k=1}^{p} \hat{\beta}_i x_{ik}$
$e_i = y_i - (\hat{\alpha} + \sum_{k=1}^{p} \hat{\beta}_i x_{ik})$
e_1
e_2
e_3
e_4
e_5
$\cdots$
e_n
$e_1^{*(1)}, e_2^{*(1)}, e_3^{*(1)}, \cdots, e_n^{*(1)}$
$e_1^{*(2)}, e_2^{*(2)}, e_3^{*(2)}, \cdots, e_n^{*(2)}$
$\cdots\cdots\cdots$
$e_1^{*(B)}, e_2^{*(B)}, e_3^{*(B)}, \cdots, e_n^{*(B)}$
$\hat{y}_i^{*(b)} = \hat{\alpha} + \sum_{k=1}^{p} \hat{\beta}_i x_{ik} + e_i^{*(b)}$
$b = 1,2, \cdots, B.$
$i = 1,2, \cdots, n$
$(x_{11}, x_{12}, \cdots, x_{1p}, \hat{y}_1^{*(b)})$
$(x_{21}, x_{22}, \cdots, x_{2p}, \hat{y}_2^{*(b)})$
$(x_{31}, x_{32}, \cdots, x_{3p}, \hat{y}_3^{*(b)})$
$(x_{41}, x_{42}, \cdots, x_{4p}, \hat{y}_4^{*(b)})$
$\cdots$
$(x_{n1}, x_{n2}, \cdots, x_{np}, \hat{y}_n^{*(b)})$

3.3.1 Fitted Model

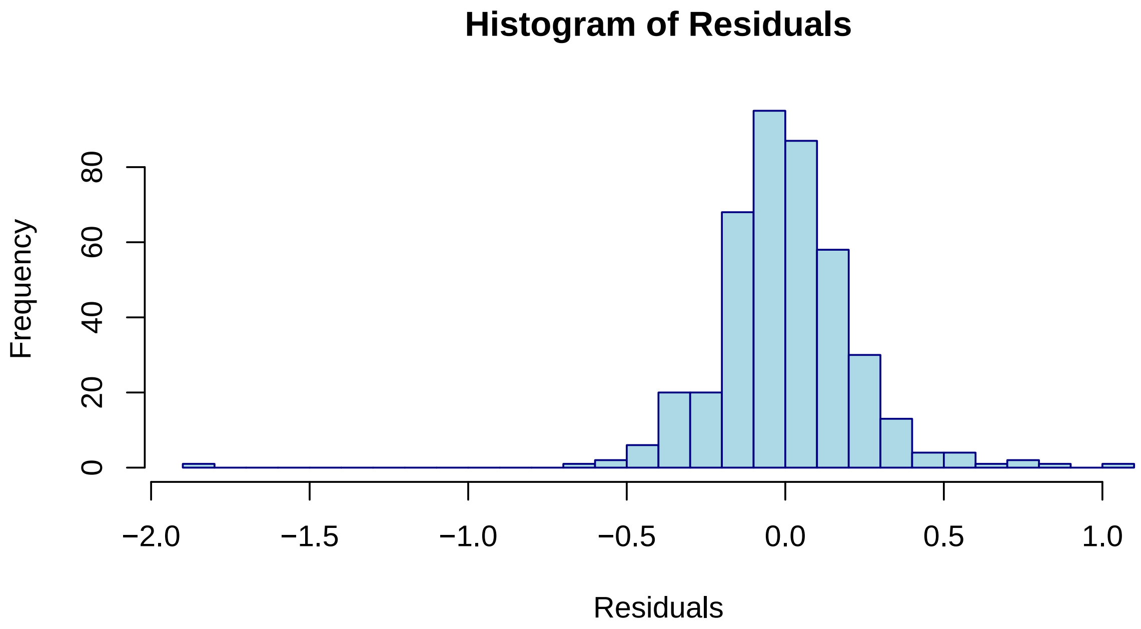
Assume that the fitted regression model is given by

$$
\begin{array}{rcl}
y_1 & = & \hat{\beta}_0 + \hat{\beta}_1 x_{11} + \hat{\beta}_2 x_{12} + \cdots + \hat{\beta}_k x_{1k} + e_1 \\
y_2 & = & \hat{\beta}_0 + \hat{\beta}_1 x_{21} + \hat{\beta}_2 x_{22} + \cdots + \hat{\beta}_k x_{2k} + e_2 \\
y_3 & = & \hat{\beta}_0 + \hat{\beta}_1 x_{31} + \hat{\beta}_2 x_{32} + \cdots + \hat{\beta}_k x_{3k} + e_3 \\
\vdots & \vdots & \vdots \\
y_n & = & \hat{\beta}_0 + \hat{\beta}_1 x_{n1} + \hat{\beta}_2 x_{n2} + \cdots + \hat{\beta}_k x_{nk} + e_n
\end{array}
$$

where $\{e_1, e_2, \cdots, e_n\}$ is the set of residuals obtained from the final model. $\{x_{i1}, x_{i2}, \cdots, x_{ik}\}$ is the i-th record from the data, and $\{\hat{\beta}_0, \hat{\beta}_1, \cdots, \hat{\beta}_k\}$ are the estimated regression coefficients based on the original random sample.

The distribution of the residuals is depicted in the following histogram.

```
hist(sort(log.price$residuals),n=40,
     xlab="Residuals",
     col = "lightblue",
     border="navy",
     main = "Histogram of Residuals")
```



The above histogram reveals the same information as we saw in the residual plot in the last note: (1) one outlier; (2). The distribution is slightly skewed to the right.

3.3.2 Residual Bootstrap Samples

The residual bootstrap sample of y is defined in the following:

- Take a **bootstrap sample** from the set of residuals $\{e_1, e_2, \cdots, e_n\}$, denoted by $\{e_1^*, e_2^*, \cdots, e_n^*\}$.
- The residual bootstrap sample of $\{y_1^*, y_2^*, \cdots, y_n^*\}$ is defined by

$$\begin{array}{ccc}
y_1^* & = & \hat\beta_0 + \hat\beta_1 x_{11} + \hat\beta_2 x_{12} + \cdots + \hat\beta_k x_{1k} + e_1^* \\
y_2^* & = & \hat\beta_0 + \hat\beta_1 x_{21} + \hat\beta_2 x_{22} + \cdots + \hat\beta_k x_{2k} + e_2^* \\
y_3^* & = & \hat\beta_0 + \hat\beta_1 x_{31} + \hat\beta_2 x_{32} + \cdots + \hat\beta_k x_{3k} + e_3^* \\
\vdots & \vdots & \vdots \\
y_n^* & = & \hat\beta_0 + \hat\beta_1 x_{n1} + \hat\beta_2 x_{n2} + \cdots + \hat\beta_k x_{nk} + e_n^*
\end{array}$$

The above definition implies that the residual bootstrap is equal to the **fitted value + bootstrap residuals**.

- The resulting **residual bootstrap sample** is given by

$$\begin{array}{ccccc}
y_1^* & x_{11} & x_{12} & \cdots & x_{1k} \\
y_2^* & x_{21} & x_{22} & \cdots & x_{2k} \\
y_3^* & x_{31} & x_{32} & \cdots & x_{3k} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
y_n^* & x_{n1} & x_{n2} & \cdots & x_{nk}
\end{array}$$

- We fit the final model to the **residual bootstrap sample** and denote the bootstrap estimates of regression coefficients in the following

$$\{\hat{\beta}_0^*, \hat{\beta}_1^*, \cdots, \hat{\beta}_k^*\}$$

- Repeat the above steps B times, we obtain the following bootstrap estimates

$$\begin{array}{cccc}
\hat{\beta}_0^{1*} & \hat{\beta}_1^{1*} & \cdots & \hat{\beta}_k^{1*} \\
\hat{\beta}_0^{2*} & \hat{\beta}_1^{2*} & \cdots & \hat{\beta}_k^{2*} \\
\hat{\beta}_0^{3*} & \hat{\beta}_1^{3*} & \cdots & \hat{\beta}_k^{3*} \\
\vdots & \vdots & \vdots & \vdots \\
\hat{\beta}_0^{b*} & \hat{\beta}_1^{b*} & \cdots & \hat{\beta}_k^{b*} \\
\vdots & \vdots & \vdots & \vdots \\
\hat{\beta}_0^{B*} & \hat{\beta}_1^{B*} & \cdots & \hat{\beta}_k^{B*}
\end{array}$$

The residual bootstrap confidence intervals of regression coefficients can be estimated from the above bootstrap coefficients.

3.3.3 Implementation of Residual Bootstrap Regression

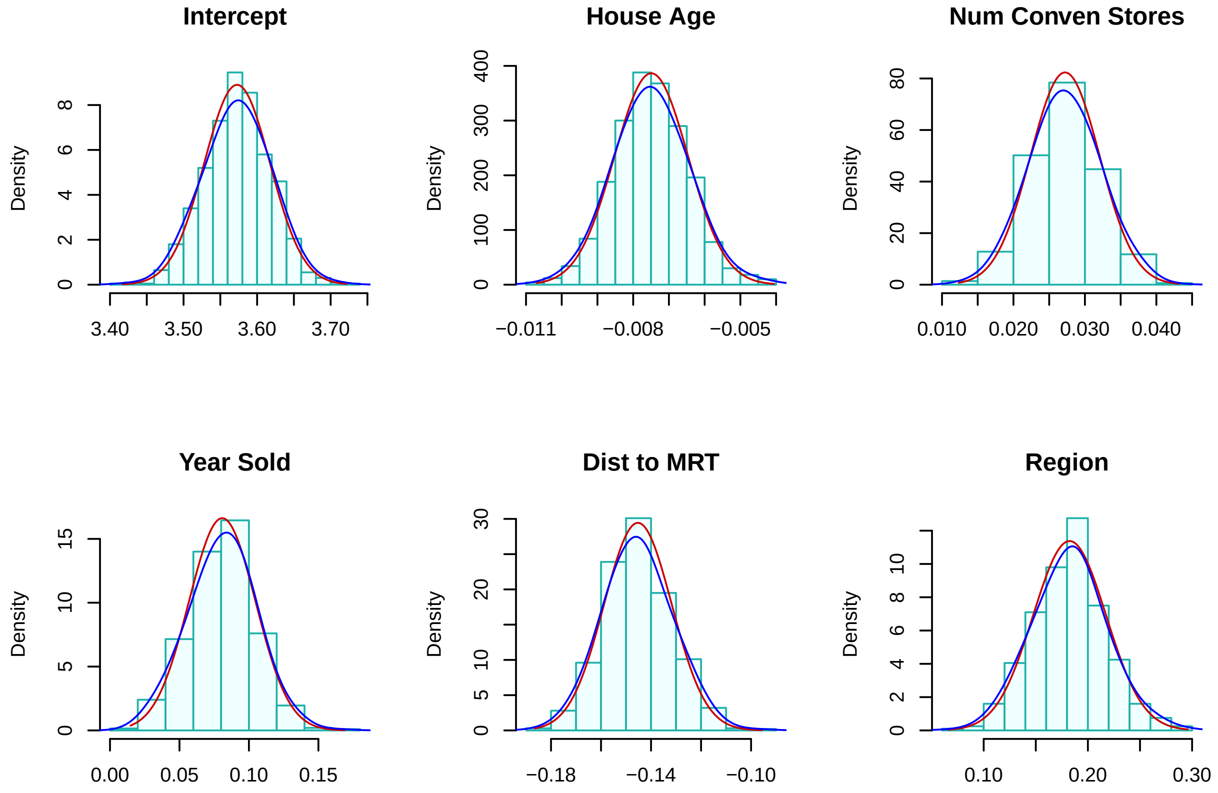
The following code generates bootstrap confidence intervals of regression coefficients.

```r
## Final model
log.price <- lm(log(PriceUnitArea) ~ HouseAge + NumConvenStores + sale.year +
                 Dist2MRT.kilo  + geo.group, data = final.data)
model.resid = log.price$residuals
##
B=1000
num.p = dim(model.matrix(log.price))[2]   # number of parameters
samp.n = dim(model.matrix(log.price))[1]  # sample size
btr.mtrx = matrix(rep(0,6*B), ncol=num.p) # zero matrix to store boot coefs
for (i in 1:B){
  ## Bootstrap response values
  bt.lg.price = log.price$fitted.values +
        sample(log.price$residuals, samp.n, replace = TRUE)  # bootstrap residuals
  # replace PriceUnitArea with bootstrap log price
  final.data$bt.lg.price =  bt.lg.price   #  send the boot response to the data
  btr.model = lm(bt.lg.price ~ HouseAge + NumConvenStores + sale.year +
                 Dist2MRT.kilo  + geo.group, data = final.data)   # b
  btr.mtrx[i,]=btr.model$coefficients
}
```

Next, I make histograms of the residual bootstrap estimates of the regression coefficients.

```r
boot.hist = function(bt.coef.mtrx, var.id, var.nm){
  ## bt.coef.mtrx = matrix for storing bootstrap estimates of coefficients
  ## var.id = variable ID (1, 2, ..., k+1)
  ## var.nm = variable name on the hist title, must be the string in the double quotes
  ## Bootstrap sampling distribution of the estimated coefficients
  x1.1 <- seq(min(bt.coef.mtrx[,var.id]), max(bt.coef.mtrx[,var.id]), length=300 )
  y1.1 <- dnorm(x1.1, mean(bt.coef.mtrx[,var.id]), sd(bt.coef.mtrx[,var.id]))
  # height of the histogram - use it to make a nice-looking histogram.
  highestbar = max(hist(bt.coef.mtrx[,var.id], plot = FALSE)$density)
  ylimit <- max(c(y1.1,highestbar))
  hist(bt.coef.mtrx[,var.id], probability = TRUE, main = var.nm, xlab="",
       col = "azure1",ylim=c(0,ylimit), border="lightseagreen")
  lines(x = x1.1, y = y1.1, col = "red3")        # normal density curve
  lines(density(bt.coef.mtrx[,var.id], adjust=2), col="blue")     # loess curve
}
```

```
par(mfrow=c(2,3))  # histograms of bootstrap coefs
boot.hist(bt.coef.mtrx=btr.mtrx, var.id=1, var.nm ="Intercept" )
boot.hist(bt.coef.mtrx=btr.mtrx, var.id=2, var.nm ="House Age" )
boot.hist(bt.coef.mtrx=btr.mtrx, var.id=3, var.nm ="Num Conven Stores" )
boot.hist(bt.coef.mtrx=btr.mtrx, var.id=4, var.nm ="Year Sold" )
boot.hist(bt.coef.mtrx=btr.mtrx, var.id=5, var.nm ="Dist to MRT" )
boot.hist(bt.coef.mtrx=btr.mtrx, var.id=6, var.nm ="Region" )
```



The residual bootstrap sampling distributions of each estimated regression coefficient. The normal and LOESS curves are close to each other. This also indicated that the inference of the significance of variables based on p-values and residual bootstrap will yield the same results.

The 95% residual bootstrap confidence intervals are given in the following

```
#
num.p = dim(coef.mtrx)[2]   # number of parameters
btr.ci = NULL
btr.wd = NULL
for (i in 1:num.p){
  lci.025 = round(quantile(btr.mtrx[, i], 0.025, type = 2),8)
  uci.975 = round(quantile(btr.mtrx[, i],0.975, type = 2 ),8)
  btr.wd[i] = uci.975 - lci.025
  btr.ci[i] = paste("[", round(lci.025,4),", ", round(uci.975,4),"]")
}
#as.data.frame(btc.ci)
kable(as.data.frame(cbind(formatC(cmtrx,4,format="f"), btr.ci.95=btr.ci)),
      caption = "Regression Coefficient Matrix with 95% Residual Bootstrap CI")
```

Table 3: Regression Coefficient Matrix with 95% Residual Bootstrap CI

	Estimate	Std. Error	t value	Pr(>|t|)	btr.ci.95
(Intercept)	3.5724	0.0443	80.5990	0.0000	[3.4873 , 3.6579]
HouseAge	-0.0076	0.0010	-7.4985	0.0000	[-0.0095 , -0.0054]
NumConvenStores	0.0275	0.0049	5.5987	0.0000	[0.0178 , 0.0371]
sale.year2013	0.0806	0.0244	3.2963	0.0011	[0.0324 , 0.129]
Dist2MRT.kilo	-0.1445	0.0138	-10.5068	0.0000	[-0.1714 , -0.1184]
geo.groupTRUE	0.1826	0.0347	5.2596	0.0000	[0.1146 , 0.2572]

As expected, the residual bootstrap confidence intervals yield the same results as p-values do. This is because the sample size is large enough so that the sampling distributions of estimated coefficients have sufficiently good approximations of normal distributions.

3.3.4 Combining All Inferential Statistics

Finally, we put all inferential statistics in a single table so we can compare these results.

```
kable(as.data.frame(cbind(formatC(cmtrx[,-3],4,format="f"), btc.ci.95=btc.ci,btr.ci.95=btr.ci)),
      caption="Final Combined Inferential Statistics: p-values and Bootstrap CIs")
```

Table 4: Final Combined Inferential Statistics: p-values and Bootstrap CIs

	Estimate	Std. Error	Pr(>|t|)	btc.ci.95	btr.ci.95
(Intercept)	3.5724	0.0443	0.0000	[3.4833 , 3.6671]	[3.4873 , 3.6579]
HouseAge	-0.0076	0.0010	0.0000	[-0.0096 , -0.0056]	[-0.0095 , -0.0054]
NumConvenStores	0.0275	0.0049	0.0000	[0.017 , 0.0379]	[0.0178 , 0.0371]
sale.year2013	0.0806	0.0244	0.0011	[0.0365 , 0.1218]	[0.0324 , 0.129]
Dist2MRT.kilo	-0.1445	0.0138	0.0000	[-0.1744 , -0.1149]	[-0.1714 , -0.1184]
geo.groupTRUE	0.1826	0.0347	0.0000	[0.1133 , 0.2507]	[0.1146 , 0.2572]

The above table shows that

- All three methods yield the same results in terms of the significance of individual explanatory variables. The reason is that the final working model does not have a serious violation of the model assumption.

```
kable(round(cbind(btc.wd, btr.wd),4), caption="width of the two bootstrap confidence intervals")
```

Table 5: width of the two bootstrap confidence intervals

btc.wd	btr.wd
0.1838	0.1706

btc.wd	btr.wd
0.0040	0.0041
0.0209	0.0192
0.0853	0.0966
0.0595	0.0530
0.1373	0.1426

We see from the above table that the widths of residual-bootstrap and case-bootstrap confidence intervals are similar to each other.

3.4 Discussions on Model Reporting

We have used different methods to perform regression analysis. The inferential results of all three major methods yield similar results. We chose the parametric model to report.

```r
kable(cmtrx, caption = "Inferential Statistics of Final Model")
```

Table 6: Inferential Statistics of Final Model

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.5724282	0.0443235	80.599030	0.0000000
HouseAge	-0.0075712	0.0010097	-7.498507	0.0000000
NumConvenStores	0.0274872	0.0049096	5.598667	0.0000000
sale.year2013	0.0805519	0.0244373	3.296272	0.0010655
Dist2MRT.kilo	-0.1445122	0.0137541	-10.506820	0.0000000
geo.groupTRUE	0.1825871	0.0347151	5.259583	0.0000002

Regression coefficients explain the correlation between log price and the corresponding explanatory variables. As an example, We explain the estimated regression coefficient of 0.1826. Let p_{TRUE} be the mean price of a house in the region `TRUE` and p_{FALSE} be the mean price of houses in the region `FALSE`. Then

$$\log(p_{TRUE}) - \log(p_{FALSE}) = 0.1826 \to \log(p_{TRUE}/p_{FALSE}) = 0.1826 \to p_{TRUE} = 1.20p_{FALSE}$$

We re-express the above equation can be re-written as

$$p_{TRUE} - p_{FALSE} = 0.206p_{FALSE} \to \frac{p_{TRUE} - p_{FALSE}}{p_{FALSE}} = 0.20 = 20\%.$$

That is, the average house sales price in the `TRUE` region (top right corner on the map) is about 20% higher than that in the `FALSE` region. We can similarly interpret other regression coefficients.

4 Some Concluding Remarks

Let Data Speak for Itself Truthfully and Understandably

4.1 Key Elements in Storytelling

Data storytelling is an effective way of communicating insights. It involves constructing a clear and relevant narrative in context, analyzing and visualizing data, and presenting findings effectively and persuasively. Data storytelling is an essential skill for statisticians.

Effective data storytelling involves three core elements: data, narrative, and visuals.

Data is the foundation of every data story.

Narrative is the structure of the data story. It's not just about the words or context that are used to explain the information extracted from data, it is about how to organize the information into a meaningful and engaging story.

Visuals are the scenes of the data story. When the data is complex, visualizing it in data charts helps the audience to **see** the **information** they might not otherwise see.

4.2 Title and Heading in Statistics Report

Although headings and titles are similar, they are distinct: A title leads the entire document and captures its content in one or two phrases; a heading leads only a chapter or section and captures only the content of that chapter or section.

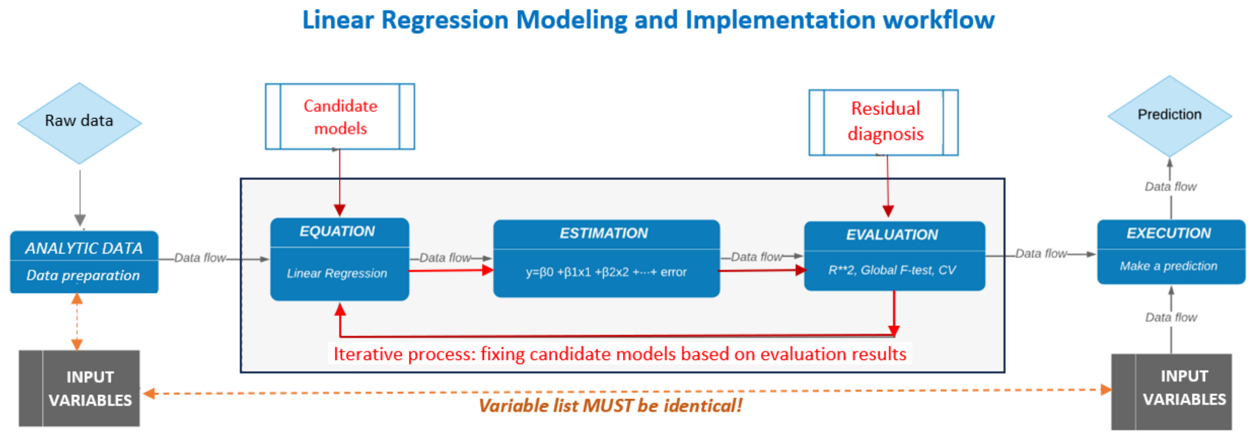
- A good **title** should be
  - descriptive,
  - Direct
  - Accurate
  - appropriate
  - concise
  - precise
  - unique and
  - should **not** be misleading.
- A good **heading** should be
  - Headings should be as long as it takes to communicate the content of the sections they head.
  - Higher-level headings often make do with a single word (e.g. **Introduction** or **Methods**), while lower-level headings are often longer.
  - Headings should be as descriptive as possible.
  - Headings should be unique.

4.3 Regression Modeling and Reporting

There are different approaches to regression modeling and reporting. We summarized the modeling process and techniques learned and suggested guidelines for reporting.

4.3.1 Modeling Workflow

Linear regression workflow is summarized in the following chart



The specific model diagnostic methods and remedies to be included in the analysis are

- residual analysis - normality, constant variance, linearity of the regression equation, etc.
- multicollinearity - variance inflation factor
- potential transformation of the response variable - conventional and Box-cox methods
- variable selection - expert-assisted variable selection (focusing on practical importance) and automatic variable selection (focusing on statistical significance).
- non-parametric robust estimation of regression coefficients - two types of bootstrap methods.

4.3.2 Reporting Guidelines

We have discussed the basic components of a statistics report. The following book chapter summarizes methods of writing and organizing statistical reports.

https://instruction.bus.wisc.edu/jfrees/jfreesbooks/Regression%20Modeling/BookWebDec2010/Writing/Chap20FreesRegBook03May2009.pdf

We have adopted the major components for our project reporting. Please review this document and pay attention to the sections related to how to report tables and figures.