AMS 572 Project Hypothesis 2

Nicholas Christophides

2024-12-04

Load the necessary packages to access the chosen data. Then initialize the data.

```
library(dplyr); library(boot); library(NHANES); library(nortest);
```

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

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

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

```
data("NHANES")
```

We observe that there are duplicate observations in the data set. This can lead to skewed results, so we wish to delete all duplicate observations. This can easily be done by using the subset() function in R.

```
NHANES=subset(NHANES, !duplicated(NHANES$ID))
```

The data now has no duplicate observations. This reduced the size of our data by over 3,000 observations. If we had not deleted these duplicates, there would be a significant chance that the results would have been skewed.

We now wish to filter out individuals below the age of 18. This is because according to the R documentation on this data set, individuals below the age of 18 were not asked to record responses for the variables we are concerned with. Show the new size of the data set after performing the filtration.

```
NHANES=subset(NHANES, Age>=18)
dim(NHANES)
```

```
## [1] 4835   76
```

The data set now has 4,835 observations. We have removed 5,165 observations by cleaning and filtering the data. This allows us to receive more meaningful results. Luckily, we still have a large number of observations.

For our hypothesis, let us create a data frame containing only relevant data.

```
dataframe=data.frame(
  AlcoholYear=NHANES$AlcoholYear,
  HardDrugs=NHANES$HardDrugs,
  HouseholdIncome=NHANES$HHIncomeMid
  )
```

Replace Yes and No in HardDrugs with a binary variable. This will allow us to fit a linear model more easily.

```
dataframe$HardDrugs=gsub("Yes", 1, dataframe$HardDrugs)
dataframe$HardDrugs=gsub("No", 0, dataframe$HardDrugs)
```

Preview the new data frame.

```
head(dataframe)
```

```
##   AlcoholYear HardDrugs HouseholdIncome
## 1           0         1           30000
## 2          20         1           40000
## 3          52         0           87500
## 4         100         0           30000
## 5         104         1          100000
## 6         364         1           70000
```

We should deal with missing data in any column by removing it. This is appropriate because we do not have any reason as to why some data is left unanswered. Due to this, we do not know what would be the best way to impute missing data. Since we will still have a large number of observations even after we delete missing data, it is a good idea to completely remove the missing data from our data frame. Show the new size of the data frame.

```
dataframe=na.omit(dataframe)
dim(dataframe)
```

```
## [1] 2942    3
```

We have now reduced the size of our data frame by 7,058 observations for a total of 2,942 observations. This is still a large number of observations, and we no longer have to worry about missing data.

Create a function that will be used to re-sample the data along with boot(). This function takes our data frame as input as well as a vector of indices. A variable boot_data is created that is the sample taken from the observed data. A multiple linear regression model is taken for each sample drawn and the coefficient on the HardDrugs variable is returned for each one.

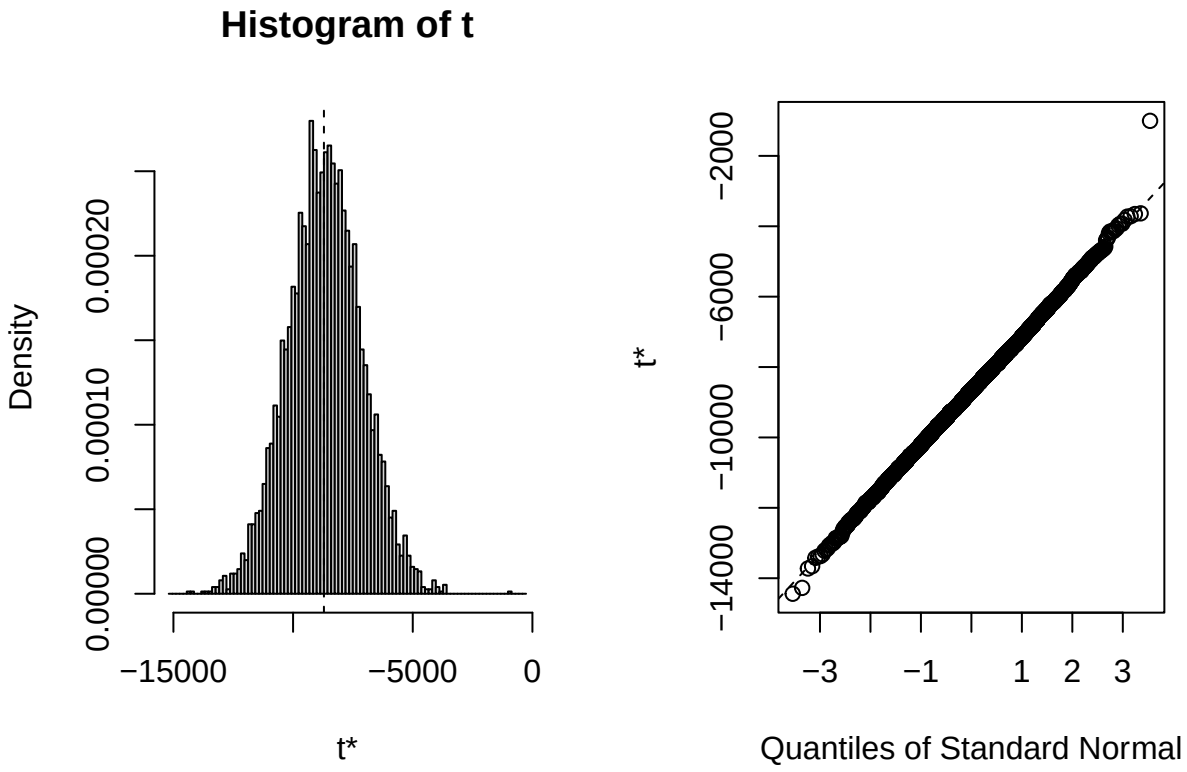
```
resampleFunc = function(dataframe, indices){
  boot_data=dataframe[indices,]
  model=lm(HouseholdIncome~HardDrugs + AlcoholYear, data=boot_data)
  return(coef(model)[2])
}
```

Perform the bootstrapping with the help of the function we defined above. Specify that we would like to receive 5,000 samples from our data. This is a large enough number to ensure precise results.

```
resampled=boot(data=dataframe, statistic=resampleFunc, R=5000)
```

View the distribution of the coefficient of HardDrugs. Also, show the residuals of the coefficients by plotting them on a Q-Q plot.

```
plot(resampled)
```



We can observe that the distribution of the 5,000 does appear to be normally distributed. The Q-Q plot seems to reinforce this assumption.

Confirm the normality assumption with an Anderson-Darling test.

```
ad.test(resampled$t)
```

```
##
##  Anderson-Darling normality test
##
## data:  resampled$t
## A = 0.17647, p-value = 0.9223
```

The test returns a p-value that is large enough to say with reasonable certainty that the distribution of the coefficients is normally distributed. Since we have now confirmed this assumption, we may move on in our hypothesis testing.

Create a 99% confidence interval for the true coefficient of HardDrugs in order to provide evidence for our hypotheses.

```r
boot.ci(resampled, type="perc", conf = 0.99)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = resampled, conf = 0.99, type = "perc")
##
## Intervals :
## Level     Percentile
## 99%   (-12796,  -4684 )
## Calculations and Intervals on Original Scale
```

Based on the results of this interval and since 0 is not within the bound of the interval, we have sufficient evidence to support the rejection of the null hypothesis that there is no statistically significant association between the use of hard drugs in one's lifetime and the household income for this individual. . . . . .

The second section of this analysis will be to take an approach where we must impute values for a category based on a given criteria and compare the results of this data with imputation to the results we recieved in the first section.

We will impute values for HardDrugs based on the entry in Alcohol12PlusYear for that individual.

Create a new data frame for this alternative testing method, including the extra variables now needed.

```r
dataframe_alt=data.frame(
  AlcoholYear=NHANES$AlcoholYear,
  Alcohol12PlusYear=NHANES$Alcohol12PlusYr,
  HardDrugs=NHANES$HardDrugs,
  HouseholdIncome=NHANES$HHIncomeMid
)
```

We will now delete missing values from Alcohol12PlusYear and HouseholdIncome, since if these values are missing, there is no meaningful imputation that we can perform. We will need these variables to have values to make a meaningful imputation for HardDrugs.

```r
dataframe_alt=subset(dataframe_alt, !is.na(Alcohol12PlusYear))
dataframe_alt=subset(dataframe_alt, !is.na(HouseholdIncome))
```

Replace Yes and No in HardDrugs with a binary variable, in order to allow our multiple linear regression fitting to run seamlessly later in the project. This is just repeating what we executed above but for our new data frame.

```r
dataframe_alt$HardDrugs=gsub("Yes", 1, dataframe_alt$HardDrugs)
dataframe_alt$HardDrugs=gsub("No", 0, dataframe_alt$HardDrugs)
```

Since we may use Alcohol12PlusYear to determine whether a person has consumed more than 12 alcoholic beverages this year, we can use that to impute a value for AlcoholYear if that is missing. We will impute a value of 6 if the value for AlcoholYear is missing. This is because we do not know an exact value to impute, but 6 is in the middle of the range that we do know the true answer lies within.

We also can observe that the if the variable Alcohol12PlusYear is No, then the likelihood of HardDrugs being No is very high. We can use this observation to impute values of 0 for HardDrugs whenever Alcohol12PlusYear is No and HardDrugs is missing a value.

When this is done remove potential remaining missing values that cannot be imputed.

```
dataframe_alt=dataframe_alt %>% mutate(
  AlcoholYear=ifelse(
    Alcohol12PlusYear=="No" & is.na(AlcoholYear), 6, AlcoholYear),
  HardDrugs=ifelse(
    Alcohol12PlusYear=="No" & is.na(HardDrugs), "0", HardDrugs)
)

dataframe_alt=na.omit(dataframe_alt)
```

We gained an extra 516 values by imputing data based on the criteria above as opposed to the original method.

```
dim(dataframe_alt)
```

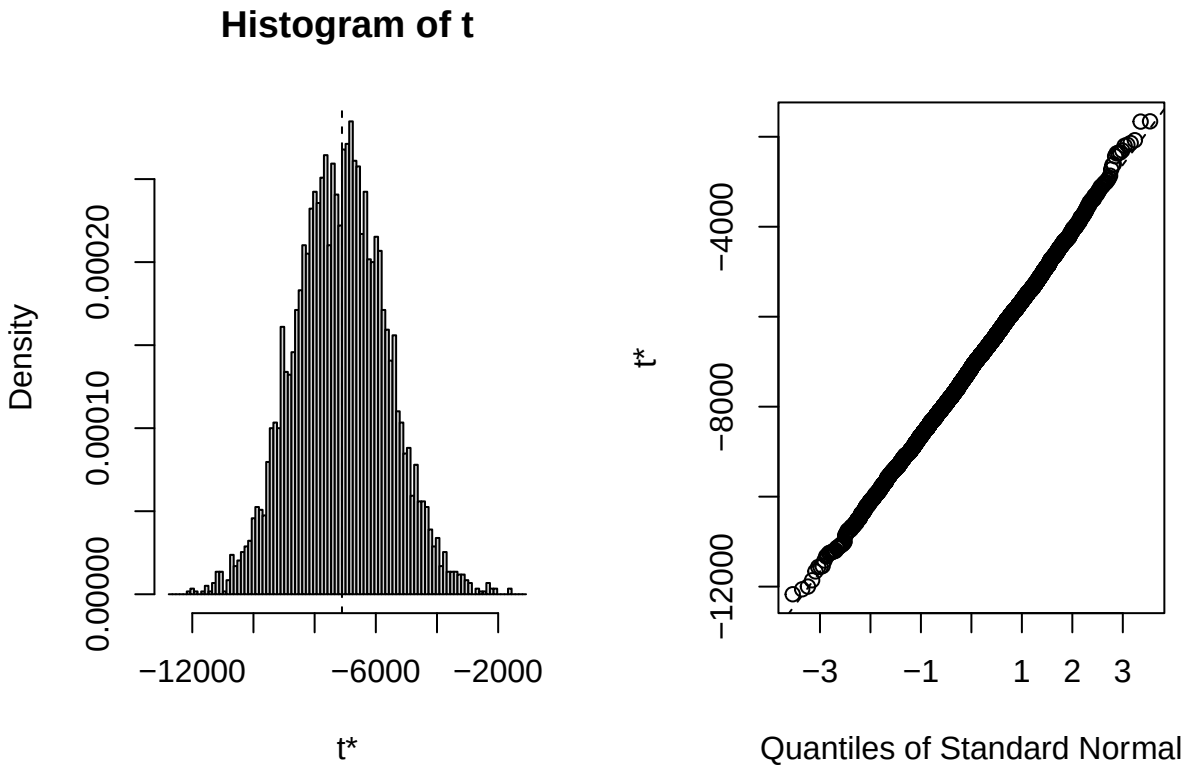
```
## [1] 3458    4
```

Redo the bootstrapping with the new data frame that contains imputed values.

```
resampleFunc_alt = function(dataframe_alt, indicies){
  boot_data=dataframe_alt[indicies,]
  model=lm(HouseholdIncome~HardDrugs + AlcoholYear, data=boot_data)
  return(coef(model)[2])
}
resampled_alt=boot(data=dataframe_alt, statistic=resampleFunc_alt, R=5000)
```

Review distribution of coefficients and check for normality using a histogram and a Q-Q plot.

```
plot(resampled_alt)
```



The distribution of these coefficients still appears somewhat normal, but the visual test leaves uncertainty. Allow us to try the Anderson-Darling test in order to get more evidence for the normality assumption.

```
ad.test(resampled_alt$t)
```

```
##
##  Anderson-Darling normality test
##
## data:  resampled_alt$t
## A = 0.37764, p-value = 0.4085
```

The results of the Anderson-Darling test still support the normality assumption of our coefficients, however the evidence here is far less strong than in the previous method. Although we have gained data points using this imputation method, we are losing certainty about the assumptions that allow us to draw conclusions about our data.

Allow us to continue with the analysis and determine a 99% confidence interval for the coefficient of Hard-Drugs.

```
boot.ci(resampled_alt, type="perc", conf = 0.99)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 5000 bootstrap replicates
##
## CALL :
```

```
## boot.ci(boot.out = resampled_alt, conf = 0.99, type = "perc")
##
## Intervals :
## Level     Percentile
## 99%   (-11064,  -3107 )
## Calculations and Intervals on Original Scale
```

The new confidence interval results in the same conclusion that we came to above, but in this case the bounds of the interval seem to be noticeably different.

Given that the normality assumption of the distribution of our coefficients is far more supported in the first method than in the second method, we recommend utilizing the results from the first method over the results from the second method.