Homework 1
MSBA 400: Statistical Foundations for Data Analytics
UID 106082225, Sarvari Pidaparty

Question 1

Review the basics of summation notation and covariance formulas. Show that:

a. $\sum_{i=1}^{N}(Y_i - \bar{Y}) = 0$
b. $\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^{N}(X_i - \bar{X})Y_i$

Answer 1

a.
To Prove: $\sum_{i=1}^{N}(Y_i - \bar{Y}) = 0$
LHS: $\sum_{i=1}^{N}(Y_i - \bar{Y}) = \sum_{i=1}^{N}(Y_i) - \sum_{i=1}^{N}\bar{Y}$

We know that: $\bar{Y} = \frac{\sum_{i=1}^{N}(Y_i)}{N}$
So: $\sum_{i=1}^{N}(Y_i) = N\bar{Y}$

Substituting this back in LHS: $N\bar{Y} - \sum_{i=1}^{N}\bar{Y} = N\bar{Y} - N\bar{Y} = 0$
Thus, LHS = RHS

b.
To Prove: $\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^{N}(X_i - \bar{X})Y_i$

LHS:
$\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y}) = \sum_{i=1}^{N}(X_i - \bar{X})(Y_i) - \sum_{i=1}^{N}(X_i - \bar{X})\bar{Y} = \sum_{i=1}^{N}(X_i - \bar{X})(Y_i) - \bar{Y}\sum_{i=1}^{N}(X_i - \bar{X})$

From a. it follows that $\sum_{i=1}^{N}(X_i - \bar{X}) = 0$, hence the entire second term in the above expression becomes 0.

So LHS $= \sum_{i=1}^{N}(X_i - \bar{X})(Y_i)$

Thus, LHS = RHS

Question 2

Define both and explain the difference between (a) the expectation of a random variable and (b) the sample average?

Answer 2

Expectation of a random variable refers to the weighted average of all the possible values that the random variable ‘X’ can take, where the weights are the probabilities of the random variable taking each of the values.

Sample Average is the mean/average of the sample space that is chosen from the population. The sample is only a small part of the population which is the whole, but sample average is used as an estimate for the population average.

While the (sample) mean is simply the average of all the values in consideration, expectation takes into account the probabilities of each of the values - it is probability-weighted. Another difference to note is that the expected value is calculated prior to the events occurring, and the average is calculated once the observations have been recorded.

Question 3

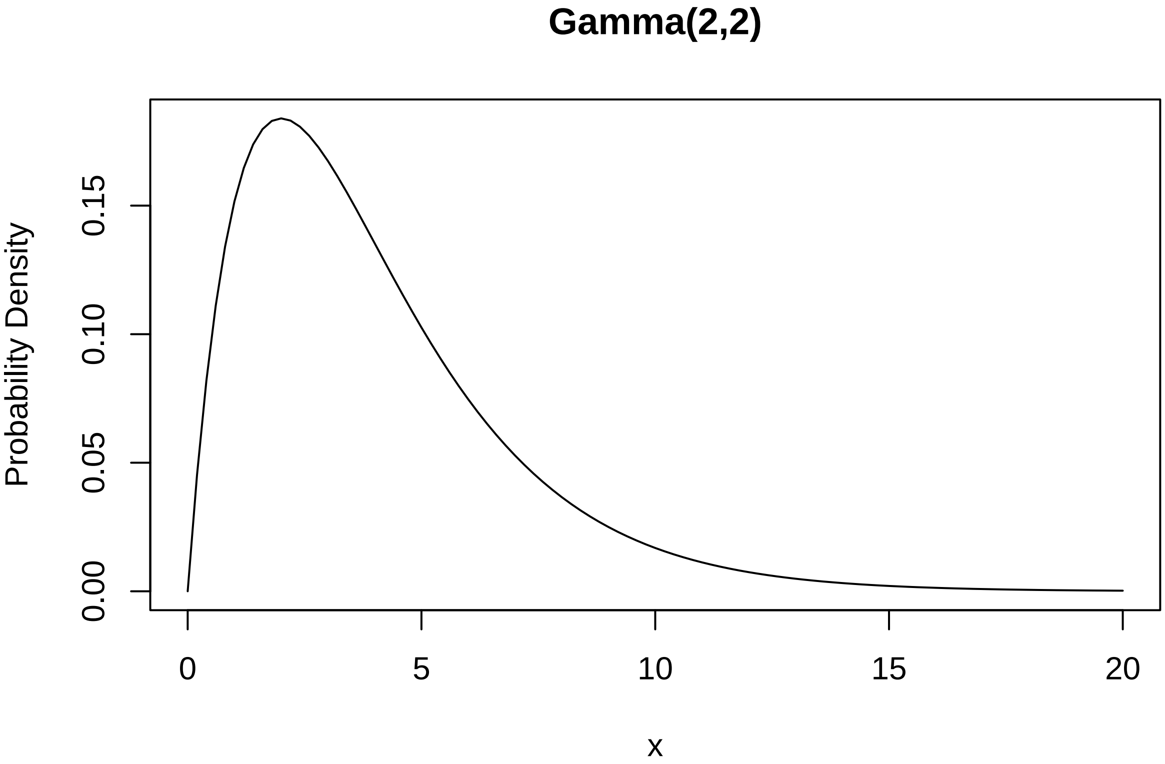
a. Describe the Central Limit Theorem as simply as you can.
b. Let $X \sim \text{Gamma}(\alpha = 2,\ \beta = 2)$. For the Gamma distribution, α is often called the "shape" parameter, β is often called the "scale" parameter, and the $\mathbb{E}[X] = \alpha\beta$. Plot the density of X and describe what you see. You may find the functions `dgamma()` or `curve()` to be helpful.
c. Let n be the number of draws from that distribution in one sample and r be the number of times we repeat the process of sampling from that distribution. Draw an iid sample of size $n = 10$ from the Gamma(2,2) distribution and calculate the sample average; call this $\bar{X}_n^{(1)}$. Repeat this process r times where $r = 1000$ so that you have $\bar{X}_n^{(1)}, \dots, \bar{X}_n^{(r)}$. Plot a histogram of these r values and describe what you see. This is the sampling distribution of $\bar{X}_{(n)}$.
d. Repeat part (c) but with $n = 100$. Be sure to produce and describe the histogram. Explain how this illustrates the CLT at work.

Answer 3

a. The Central Limit Theorem states that as we add up more and more distributions (that are more or less similar), their average starts to look more and more normally distributed. The individual distributions need not be normally distributed for this theorem to be valid.

b.

```
set.seed(0)
x <- seq(from = 0, to = 20, by = 0.01)
curve(dgamma(x, shape = 2, scale = 2), xlim = c(0,20), ylab = "Probability Density", main = "Gamma(2,2)
```



The density is high at values x = 0 to 5 and the graph peaks at about x = 2. The density decreases rapidly as the value of x increases. The graph is right-skewed.

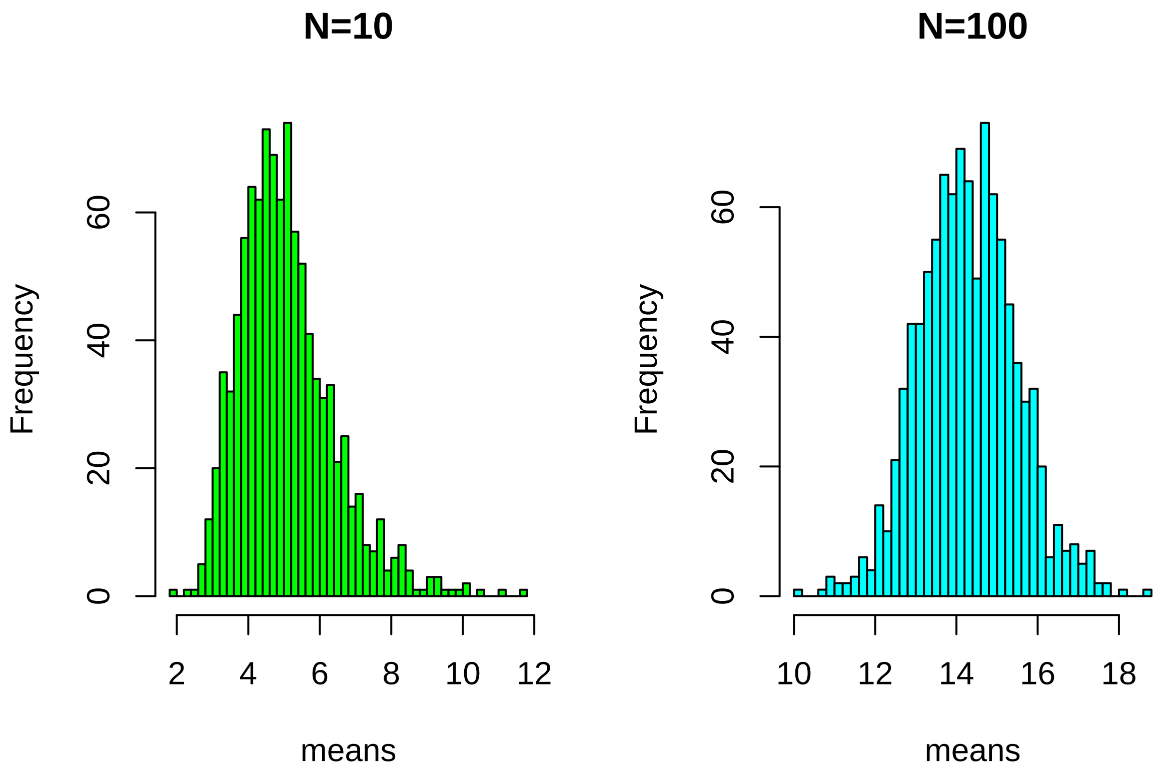
c. & d.

```
par(mfrow=c(1,2))
set.seed(0)

Nsamples = 1000
N = 10
sims = matrix(rgamma(Nsamples*N,2,2),ncol=Nsamples)
means = apply(sims,MARGIN=2,mean)
sds = apply(sims,2,sd)
means=means/(sds/sqrt(N))

hist(means,breaks=50,col="green",main="N=10")

N = 100
sims = matrix(rgamma(Nsamples*N,2,2),ncol=Nsamples)
means = apply(sims,MARGIN=2,mean)
sds = apply(sims,2,sd)
means=means/(sds/sqrt(N))
hist(means,breaks=50,col="cyan",main="N=100")
```



The histograms for both N=10 and N=100 have been produced, and we can see how as we've increased the value of N (to 100), the distribution has become much more normally distributed in comparison with the histogram with a lower N (10) value. This illustrates the Central Limit theorem at work as it describes how the summation of distributions tends to look more normally distributed as we increase the number of distributions we sum together.

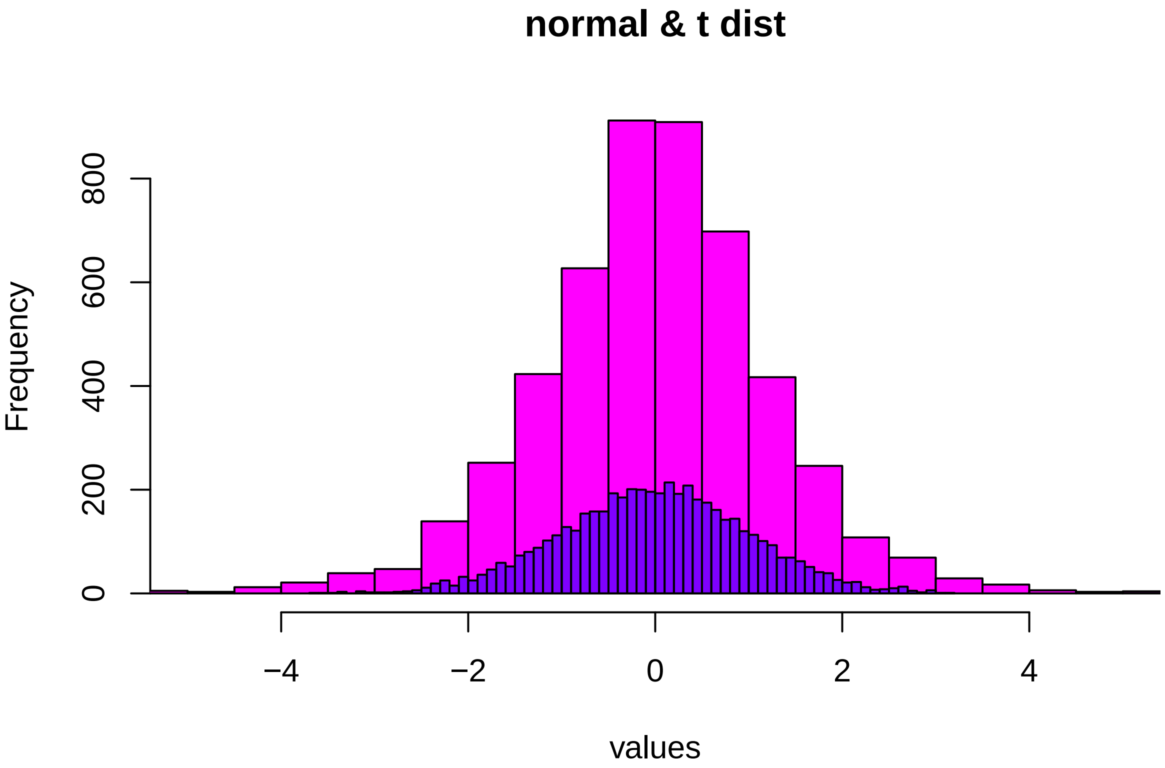
Question 4

The normal distribution is often said to have "thin tails" relative to other distributions like the *t*-distribution. Use random number generation in R to illustrate that a $\mathcal{N}(0, 1)$ distribution has much thinner tails than a *t*-distribution with 5 degrees of freedom.

A few coding hints: `rnorm()` and `rt()` are the functions in R to draw from a normal distribution and a *t*-distribution. The option `add=TRUE` for the `hist()` command can be used to overlay a second histogram on top of another histogram, and after installing the `scales` package, you can make a blue histogram 50% transparent with the option `col=scales::alpha("blue",0.5)`. Alternatively, you can put two plots side-by-side by first setting the plotting parameter with the code `par(mfrow=c(1,2))`. You can set the range of the x-axis to go from -5 to 5 with the plotting option `xlim=c(-5,5)`.

Answer 4

```
norm_dist = rnorm(5000,0,1)
t_dist = rt(5000,5)
hist(t_dist,breaks=60,col="magenta",xlim=c(-5,5),xlab = "values", main="normal & t dist")
hist(norm_dist,breaks=60,col=scales::alpha("blue",0.5),xlim=c(-5,5),add=TRUE)
```



As illustrated above, the Normal distribution has thinner tails than the t-distribution. However, t-distributions with higher degree of freedom have thinner tails than those with a lower degrees of freedom.

Question 5

a. From the Vanguard dataset, compute the standard error of the mean for the `VFIAX` index fund return.
b. For this fund, the mean and the standard error of the mean are almost exactly the same. Why is this a problem for a financial analyst who wants to assess the performance of this fund?
c. Calculate the size of the sample which would be required to reduce the standard error of the mean to 1/10th of the size of the mean return.

Answer 5

a.

```
library("DataAnalytics")
data(Vanguard)
```

```
standard_error <- function(a)
{
  sd(a)/sqrt(length(a))
}

vfiax_c = Vanguard$ticker=='VFIAX'
vfiax_mret = Vanguard[vfiax_c,]$mret
standard_error(vfiax_mret)
```

```
## [1] 0.003670128
```

```
mean(vfiax_mret)
```

```
## [1] 0.003959993
```

b. The value of standard deviation being almost equal to the mean indicates that the variation in the data set is really high, or the fund is highly "volatile", which is also considered risky to invest in. A financial analyst who would want to assess the performance of the fund would not be able to accurately estimate the returns, or predict accurately if the fund would yield appropriate returns. If we look below at the descriptive statistics, we see that the 95% confidence interval range on returns for the fund VFIAX is -0.003 to 0.011, which shows how volatile the fund is. Returns could be anywhere within this range, which is really large.

```
library(reshape2)
Vanguard_new=Vanguard[,c(1,2,5)]
V_reshaped = dcast(Vanguard_new,date~ticker,value.var="mret")
descStat(V_reshaped)
```

```
##        Mean Median    SD   IQR SE Mean 95% CI-L 95% CI-U NMissing
## VEIPX 0.009  0.012 0.037 0.043   0.002    0.005    0.013       46
## VFIAX 0.004  0.011 0.045 0.050   0.004   -0.003    0.011      198
## VGENX 0.012  0.012 0.060 0.073   0.003    0.005    0.018        0
## VGHCX 0.014  0.016 0.041 0.046   0.002    0.010    0.018        0
## VMGRX 0.010  0.013 0.072 0.073   0.005    0.000    0.021      163
## VQNPX 0.009  0.014 0.045 0.055   0.003    0.004    0.014       31
## VSMAX 0.009  0.017 0.059 0.072   0.005    0.000    0.018      198
## VTSAX 0.005  0.012 0.046 0.053   0.004   -0.003    0.012      198
## VWNFX 0.009  0.014 0.043 0.048   0.002    0.005    0.014       13
## Number of Observations =  349
```

c. We know that standard error $s_{\bar{y}} = \frac{s_y}{\sqrt(N)}$ Hence to obtain $\frac{s_{\bar{y}}}{10}$, new sample size N would need to be 100 times the original sample size. Original sample size = 151 New sample size will have to be 15100.

```
N_vfiax = nrow(Vanguard[vfiax_c,])
N_vfiax_new = 100 * N_vfiax
N_vfiax_new
```

```
## [1] 15100
```

Question 6 : Subsetting Observations

We have seen that R has very powerful capabilities to find observations with specific characteristics. The `[<select rows>,<select columns>]` notation is used to subset a data frame. For example, `cars[1:100,]` selects the first 100 observations (rows) of the `cars` data frame.

The most powerful subsetting is done by using what are called logical expressions. A logical expression is an expression that is either TRUE or FALSE. A very simple example would be

```
var = "Ford"     # 1
var == "Ford"    # 2
```

```
## [1] TRUE
```

```
var != "Ford"    # 3
```

```
## [1] FALSE
```

In statement #1, we are assigning the value of "Ford" to the R object called `var`. This makes `var` a character vector with only one element.
In statement #2, we are comparing what is stored in `var` to the string "Ford", `==` is the "equals" comparison operator. If the contents of `var` is equal to "Ford", then R will produce the result `TRUE`.
In statement #3, we use the "not equals" comparison operator. #3 will return `FALSE`.

Logical Comparison Operators

Operator	Meaning
`==`	equals
`!=`	not equal
`>`	greater than
`<`	less than
`>=`	greater than or equal to
`<=`	less than or equal to

These ideas can be extended to the vectors.

```
vec=c("BMW","Cadillac","Audi")
vec == "BMW"
```

```
## [1]  TRUE FALSE FALSE
```

```
vec != "BMW"
```

```
## [1] FALSE  TRUE  TRUE
```

What does this have to do with selecting rows in a data frame (or subsetting our data)? We can use a vector that only contains "TRUE" or "FALSE" to select rows. Let's look at a simple example.

```
library(DataAnalytics)
data(mvehicles)
cars=mvehicles[mvehicles$bodytype != "Truck",]
cars_f4 = head(cars,n=4)
cars_f4[,1:3]                                  # show only the first three columns to save space in the
```

```
##   make year    model
## 1  BMW 2011 5 Series
## 2  BMW 2011 5 Series
## 3  BMW 2011 5 Series
## 4  BMW 2011 5 Series
```

```
cars_f4[c(FALSE,TRUE,FALSE,TRUE),1:3]
```

```
##   make year    model
## 2  BMW 2011 5 Series
## 4  BMW 2011 5 Series
```

The last statement in the code block above has selected the second and fourth rows. This looks like a much more cumbersome way to do it than just `cars_f4[c(2,4),]`. However, the power comes when you try to select on the values of some of the variables in the data frame which describe which observation is which.

As we saw, if we want to select all of the Fords in the data, we can simply write

```r
fords = cars[cars$make == "Ford",]
```

We now know how this works. `cars$make == "Ford"` creates a vector with TRUE values marking any observation where `make == "Ford"` and FALSE when not.

Q6, Part A

1. Display the contents of the first 50 elements of the vector, `cars$make == "Ford"`, to verify that it is a logical vector.
2. Subset the `cars` data frame by a two step process to only the "Ford" make. That is, create the row selection logical vector in one statement and select observations from the `cars` data frame in the second.
3. How many Kia observations are there in the `cars` data frame? hint: `nrow()` tells you how many rows are in a data frame.
4. How many cars are have a price (emv) that is greater than $100,000?

We can also couple two logical expressions together using AND `&` or OR `|`. For example, if we want to select all rows with either Kia or Hyundai; we would say `cars[cars$make == "Kia" | cars$make == "Hyundai",]`.

Answer to Q6, Part A

1.

```r
library("DataAnalytics")
data(mvehicles)
cars=mvehicles[mvehicles$bodytype != "Truck",]
ford_cars = head(cars$make == "Ford", 50)
ford_cars
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## [37]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE
```

2.

```r
fords = cars$make == 'Ford'
head(cars[fords,])
```

```
##    make year    model                              style  origin bodytype
## 6  Ford 2011     Edge     Limited 4dr SUV (3.5L 6cyl 6A) America      SUV
## 7  Ford 2011     Edge Limited 4dr SUV AWD (3.5L 6cyl 6A) America      SUV
## 8  Ford 2011     Edge          SE 4dr SUV (3.5L 6cyl 6A) America      SUV
## 9  Ford 2011     Edge         SEL 4dr SUV (3.5L 6cyl 6A) America      SUV
## 10 Ford 2011     Edge     SEL 4dr SUV AWD (3.5L 6cyl 6A) America      SUV
## 29 Ford 2011 Explorer             4dr SUV (3.5L 6cyl 6A) America      SUV
##         emv seats roominess mpg  warranty appearance    luxury    sporty
## 6  35441.33     5 0.3917259  22 0.3333333  0.6205152 0.4171626 0.3805899
## 7  38441.82     5 0.3917259  20 0.3333333  0.6205152 0.4171626 0.3805899
## 8  26999.12     5 0.3917259  21 0.3333333  0.6205152 0.4171626 0.3805899
## 9  30960.75     5 0.3917259  22 0.3333333  0.6205152 0.4171626 0.3805899
## 10 32619.84     5 0.3917259  20 0.3333333  0.6205152 0.4171626 0.3805899
## 29 27579.96     7 0.4912934  20 0.3333333  0.6020550 0.4169447 0.4211533
##        speed technology sales
## 6  0.6533921  0.8125000 21424
## 7  0.6327483  0.8125000 16496
## 8  0.6533921  0.1875000 15617
## 9  0.6533921  0.6562500 23186
## 10 0.6327483  0.6562500 13485
## 29 0.6544853  0.1354167  9784
```

3.

```
kias = cars$make=='Kia'
nrow(cars[kias,])
```

```
## [1] 43
```

4.

```
emv_100k = cars$emv > 100000
nrow(cars[emv_100k,])
```

```
## [1] 37
```

Q6, Part B

1. What is the average sales for all cars made in Europe with price above $75,000?

In many data sets, there are long text fields which describe an observation. These fields are not formatted in any way and so it is difficult to use simple comparison methods to fetch observations. However, we can use the power of something called regular expressions to find any observations for which a given variable contains some character pattern. Regular expressions are very complicated to use in generality but we can get a lot of use out of a very simple expression.

The `style` variable in `cars` is a general text description variable, We can find the rows for each `style` contains any string by using the command `grepl("string",column,ignore.case=TRUE)`. For example, `grepl("hybrid",cars$style,ignore.case=TRUE)` creates a logical vector (TRUE or FALSE) to help select rows corresponding to hybrids. `cars[grepl("hybrid",cars$style,ignore.case=TRUE),]` will fetch only hybrids.

Answer to Q6, Part B

```
cars_eu = grepl("Europe",cars$origin, ignore.case=TRUE)  #found cars whose origin is Europe
cars_eu_75k = cars[cars_eu & cars$emv>75000,]  #further found cars that sell above 75000
mean(cars_eu_75k$sales)
```

```
## [1] 626.6957
```

Q6, Part C

1. How many four door vehicles are in cars?
2. How many four door sedans are in cars?

Answer to Q6, Part C

1.

```
cars_fdr = grepl("4dr",cars$style,ignore.case=TRUE)
nrow(cars[cars_fdr,])
```

```
## [1] 1105
```

2.

```
nrow(cars[cars_fdr & cars$bodytype=='Sedan',])
```

```
## [1] 432
```

Question 7 : Sales and Price relationships

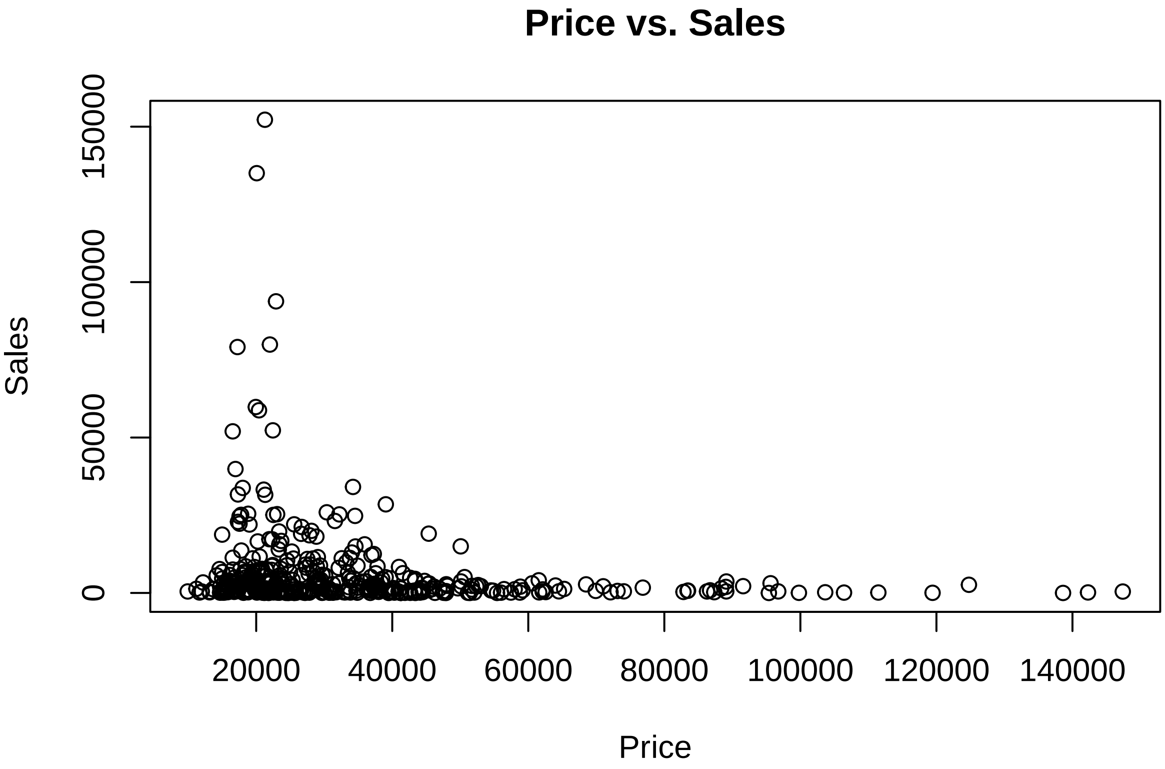
In this question, use `cars` only.

Q7, part A

Plot price (horizontal axis) vs. sales (vertical axis) for cars with bodytype == "Sedan". What is the problem with displaying the data in this manner?

Answer to Q7, part A

```
x_7A = cars[cars$bodytype=='Sedan',]$emv
y_7A = cars[cars$bodytype=='Sedan',]$sales
plot(x_7A, y_7A, xlab = "Price", ylab = "Sales", main = "Price vs. Sales")
```



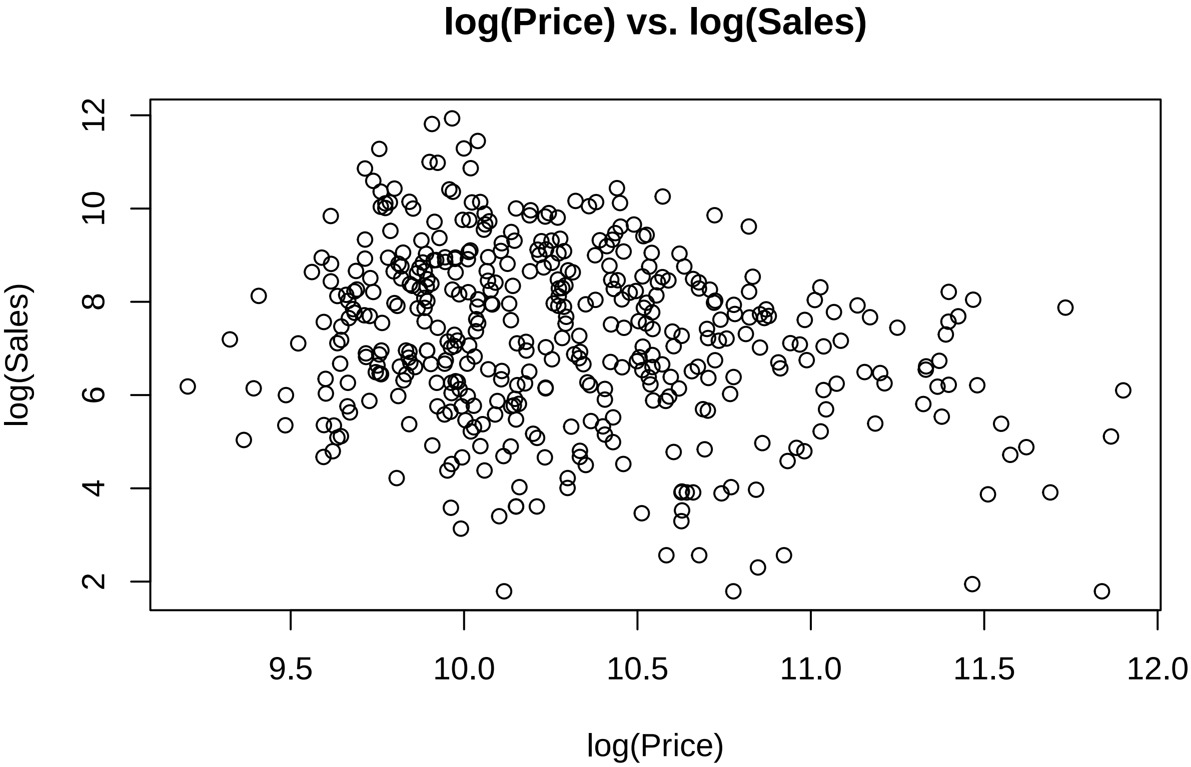
The problem with plotting the data in the above way is that it is difficult to interpret, and data is too concentrated in a certain area. There are outliers in the data as well that are spread out widely.

Q7, part B

Plot log(price) vs. log(sales) for the same subset of observations as in part 1. How has this improved the visualization of this data? Are there any disadvantages of taking the log transformation? A very similar but less "violent" transformation is the sqrt transformation. Try the sqrt transformation. Is this useful?

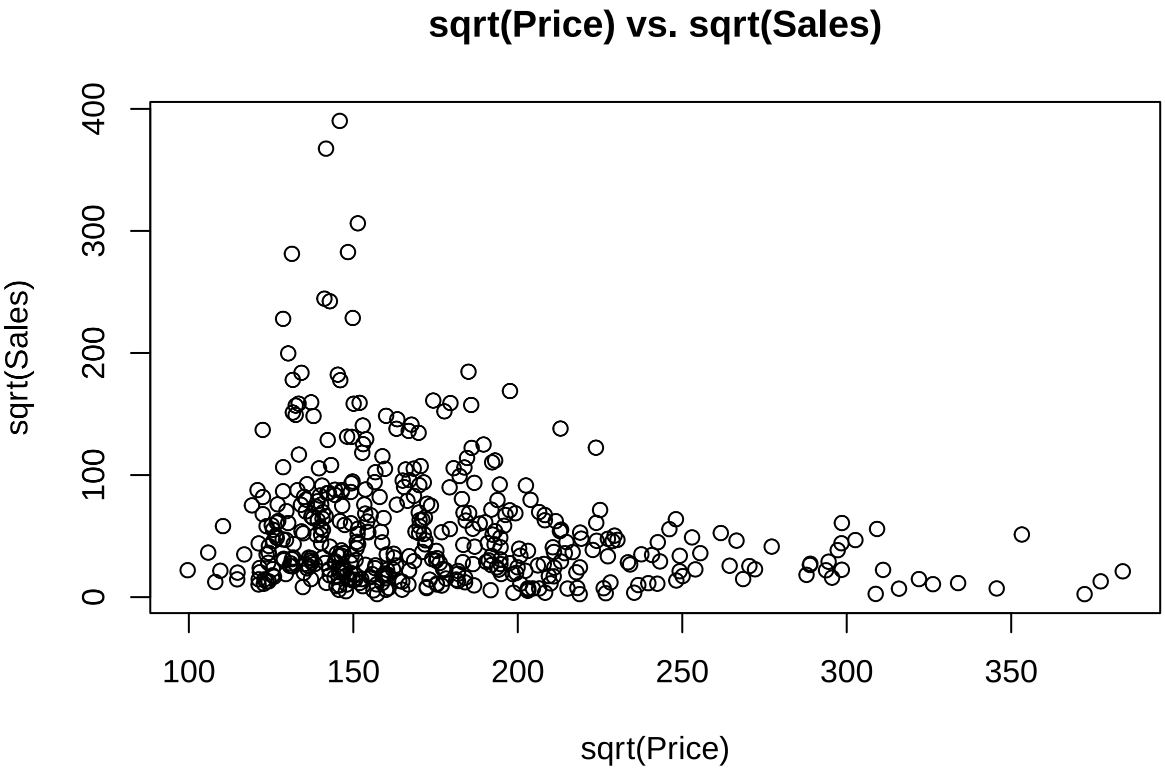
Answer to Q7, part B

```
plot(log(x_7A), log(y_7A), xlab = "log(Price)", ylab = "log(Sales)", main = "log(Price) vs. log(Sales)"
```



Applying the log transformation has brought together the data more uniformly (the outliers) and has spaced out the densely distributed values as well, making it easier for further analysis and modelling. A disadvantage of the transformation is that it may not work to reduce the skewness of a distribution that is not close to log-normal. In these cases the transformation may make the data even more skewed.

```
plot(sqrt(x_7A), sqrt(y_7A), xlab = "sqrt(Price)", ylab = "sqrt(Sales)", main = "sqrt(Price) vs. sqrt(S
```



The sqrt transformation helped decrease the area over which the data was spread out and also makes it easier to see each of the data points. The pattern is also similar to the original data plot which was generated with no transformation. The log transformation may be more useful here since it compresses the high values more aggressively.

Q7, part C

Economists will tell you that as price increase sales will decreases, all other things being equal. Does this plot support this conclusion?

Answer to Q7, part C

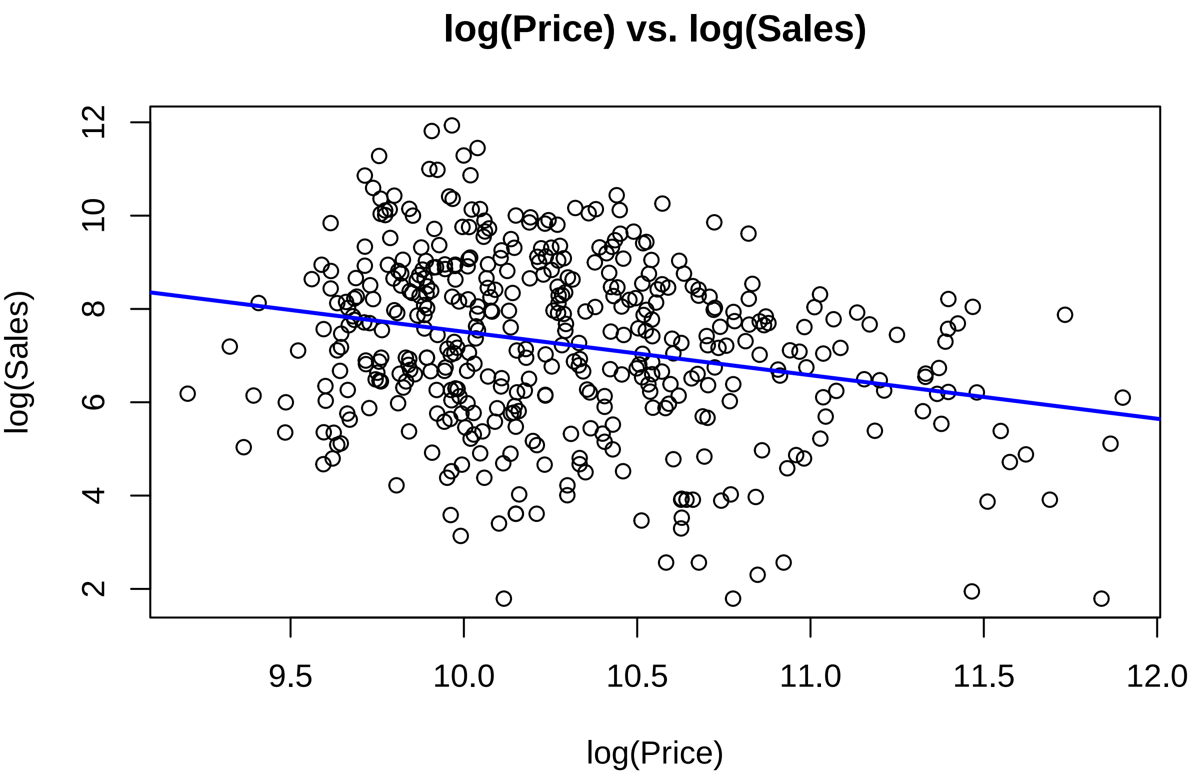
The plot does have more data points towards the left side where price is lower, but the difference is not a lot. Since these data points aren't measured keeping all other factors equal, we cannot make the conclusion in this case that price and sales are inversely correlated.

Q7, part D

Fit a regression model to this data. That is, "regress" log(sales) on log(price) (log(sales) is Y or the dependent variable). Plot the fitted line on top of the scatterplot using `abline`.

Answer to Q7, part D

```
plot(log(x_7A), log(y_7A), xlab = "log(Price)", ylab = "log(Sales)", main = "log(Price) vs. log(Sales)"
fitlm=lm(log(sales)~log(emv), data = cars[cars$bodytype=='Sedan',])
abline(fitlm$coef, lwd = 2, col = "blue")
```



Q7, part E

Predict sales for price = $45,000 using the model fit in part D). Don't forget to transform back to unit sales by using the `exp()` function.

Answer to Q7, part E

```
sales_1 = predict(fitlm, newdata = data.frame(emv = 45000))
exp(sales_1)
```

```
##        1
## 939.5983
```

–X–