Linear Regression Models, Segment 1, Topic 1: Data Generation Process Sample & Population

```
library(ggplot2)
library(dplyr)
```

```
##
## 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
```

```
# Load the mtcars dataset
file = 'data/mtcars.csv'
carData = read.csv(file, header = TRUE, row.names = 1, stringsAsFactors = FALSE)
str(carData)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : int  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : int  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : int  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : int  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: int  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: int  4 4 1 1 2 1 4 2 2 4 ...
```

```
# Convert categorical columns to represent factor levels
categorical_cols = c('cyl', 'vs', 'am', 'gear', 'carb')
carData[categorical_cols] = lapply(carData[categorical_cols], as.factor)
str(carData)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : Factor w/ 3 levels "4","6","8": 2 2 1 2 3 2 3 1 1 2 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : int  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : Factor w/ 2 levels "0","1": 1 1 2 2 1 2 1 2 2 2 ...
##  $ am  : Factor w/ 2 levels "0","1": 2 2 2 1 1 1 1 1 1 1 ...
##  $ gear: Factor w/ 3 levels "3","4","5": 2 2 2 1 1 1 1 2 2 2 ...
```

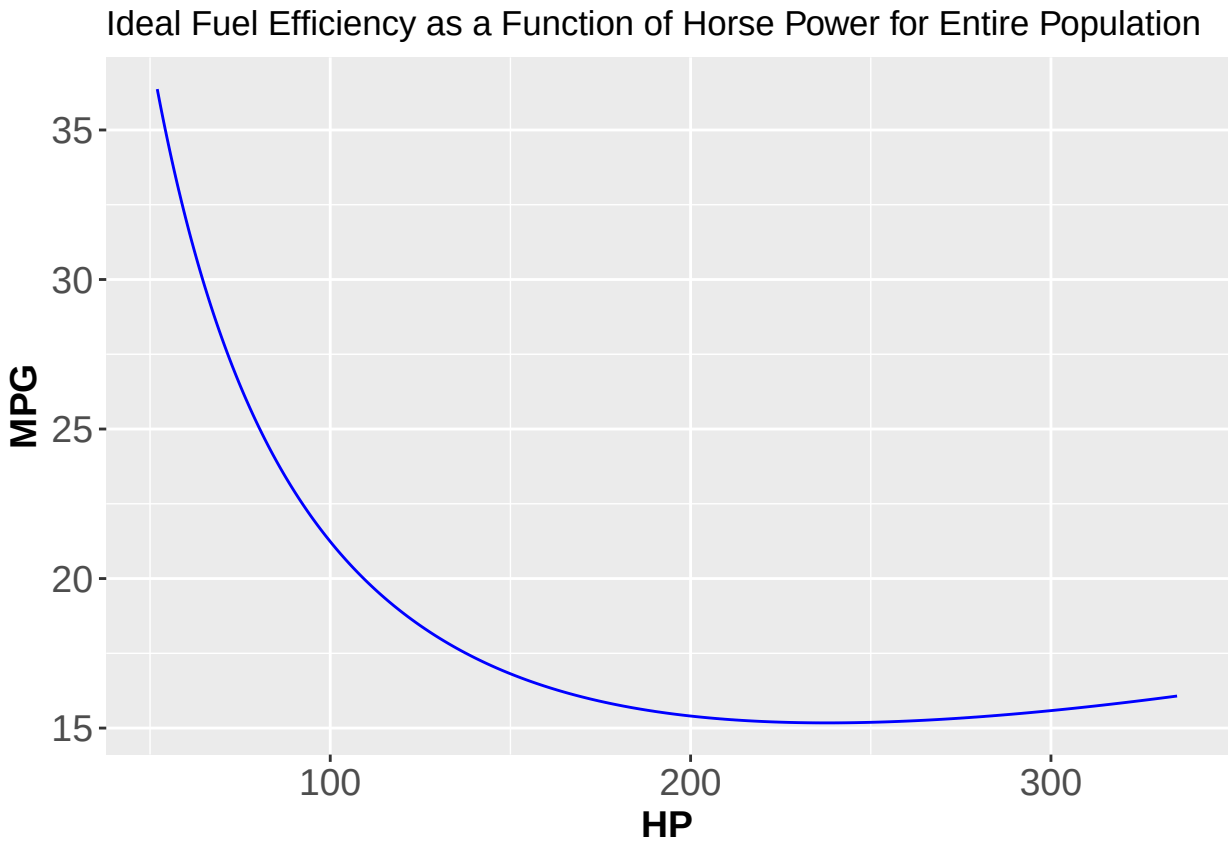
```
##  $ carb: Factor w/ 6 levels "1","2","3","4",..: 4 4 1 1 2 1 4 2 2 4 ...
```

```r
# Print the first five rows (or samples) in the data frame
head(carData, 5)
```

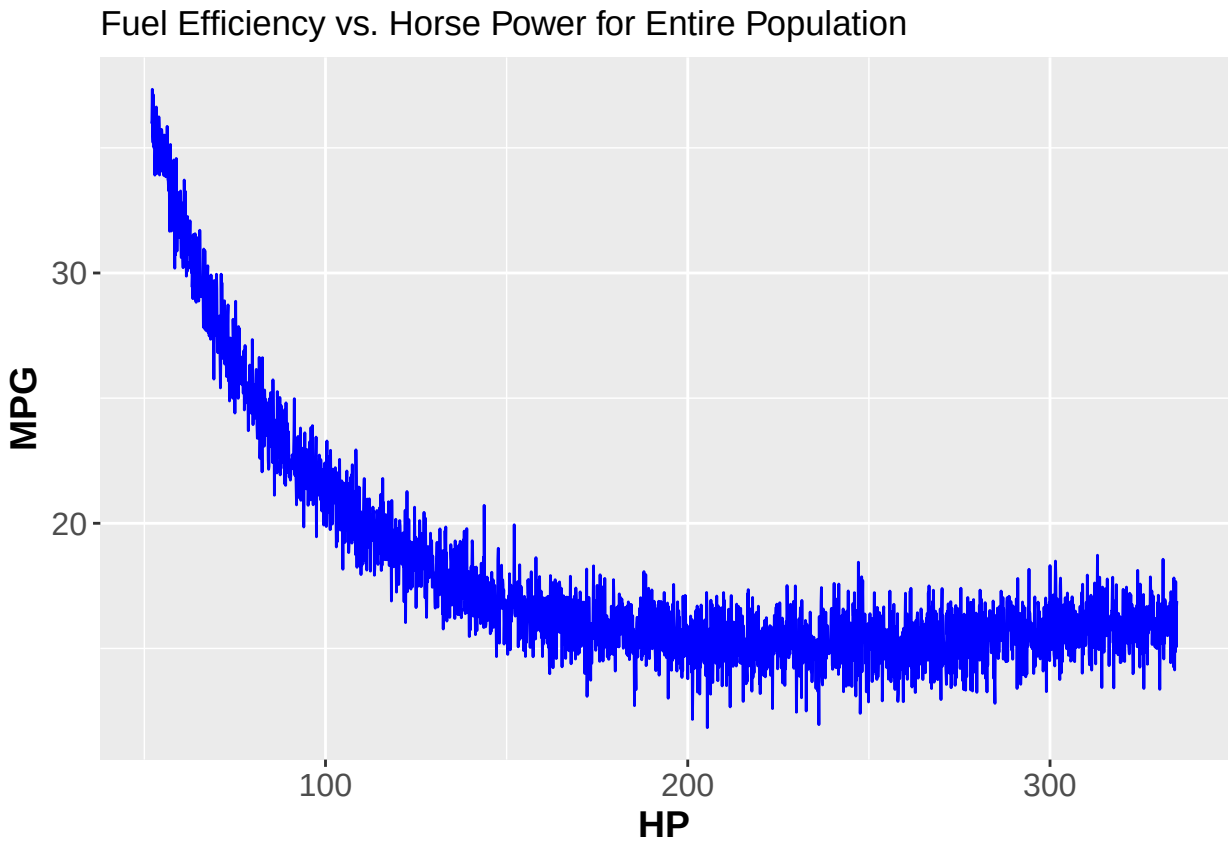
```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
```

```r
# Ideal population model for mpg vs. hp
model_ideal = nls(data = carData, mpg ~ (1 / hp) * a + b * hp, start = list(a = 1, b = 1))
calcmpgIdeal = function(hp){
  return(predict(model_ideal, list(hp = hp)))
}
```

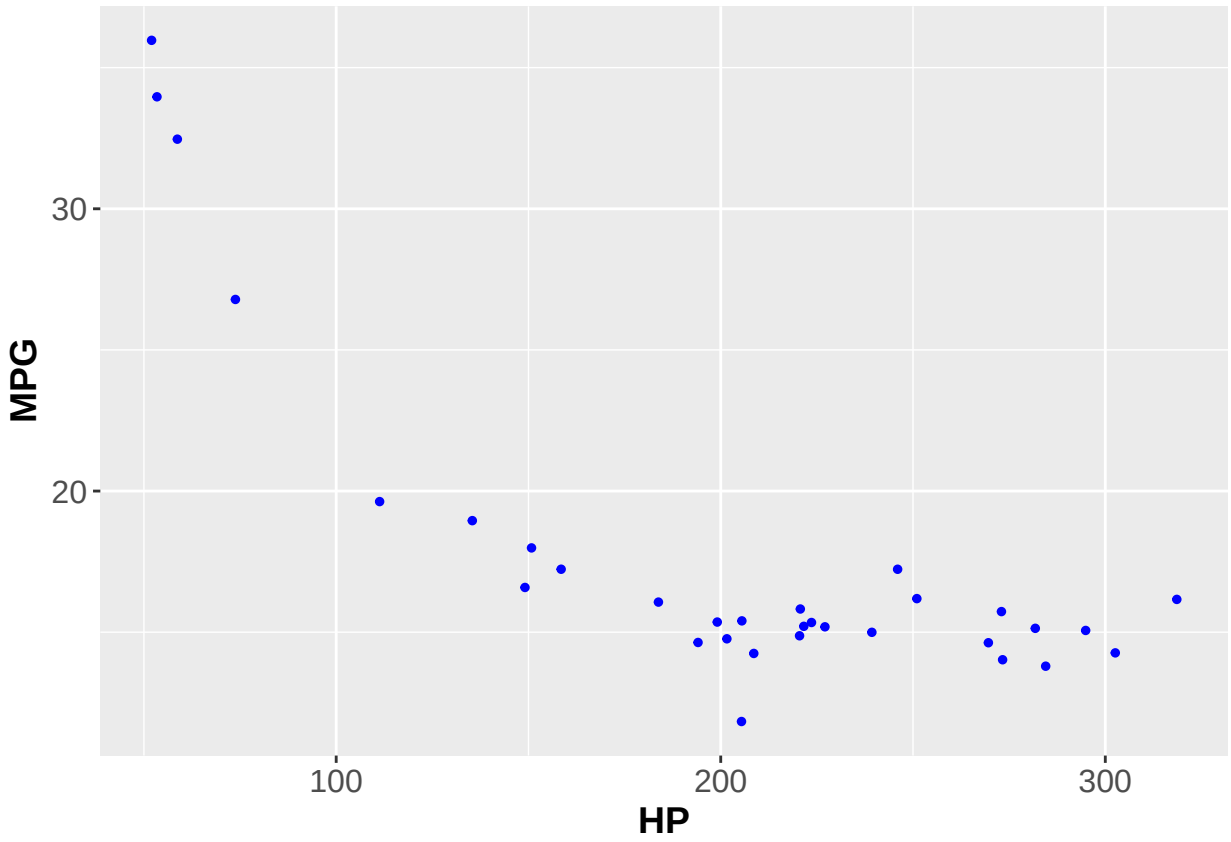
```r
# Calculate and plot ideal mpg vs. hp for entire population
hp_population = seq(min(carData$hp), max(carData$hp), by = 0.1)
mpg_population_ideal = calcmpgIdeal(hp_population)
carDataPopIdeal = data.frame(hp_population, mpg_population_ideal)
colnames(carDataPopIdeal) = c('hpPopulation', 'mpgPopulationIdeal')

ggplot(data = carDataPopIdeal, aes(x = hpPopulation, y = mpgPopulationIdeal)) +
   geom_line(color = 'blue') +
   labs(x = 'HP', y = 'MPG') +
   ggtitle("Ideal Fuel Efficiency as a Function of Horse Power for Entire Population") +
   theme(axis.text = element_text(size = 12),
   axis.text.x = element_text(size = 14),
   axis.text.y = element_text(size = 14),
   axis.title = element_text(size = 14, face = "bold"))
```




```
# Predictor noise
noise_internal = rnorm(length(hp_population), mean = 0, sd = 0.05)

# Response noise
noise_external = rnorm(length(hp_population), mean = 0, sd = 1)

# Real (noisy) population data
mpg_population_noisy = calcmpgIdeal(hp_population + noise_internal)  + noise_external
carDataPopNoisy = data.frame(hp_population, mpg_population_noisy)
colnames(carDataPopNoisy) = c('hpPopulation', 'mpgPopulationNoisy')

# Plot noisy mpg vs. hp for entire population
ggplot(data = carDataPopNoisy, aes(x = hpPopulation, y = mpgPopulationNoisy)) +
   geom_line(color = 'blue') +
   labs(x = 'HP', y = 'MPG') +
   ggtitle('Fuel Efficiency vs. Horse Power for Entire Population') +
   theme(axis.text = element_text(size = 12),
   axis.text.x = element_text(size = 12),
   axis.text.y = element_text(size = 12),
   axis.title = element_text(size = 14, face = "bold"))
```

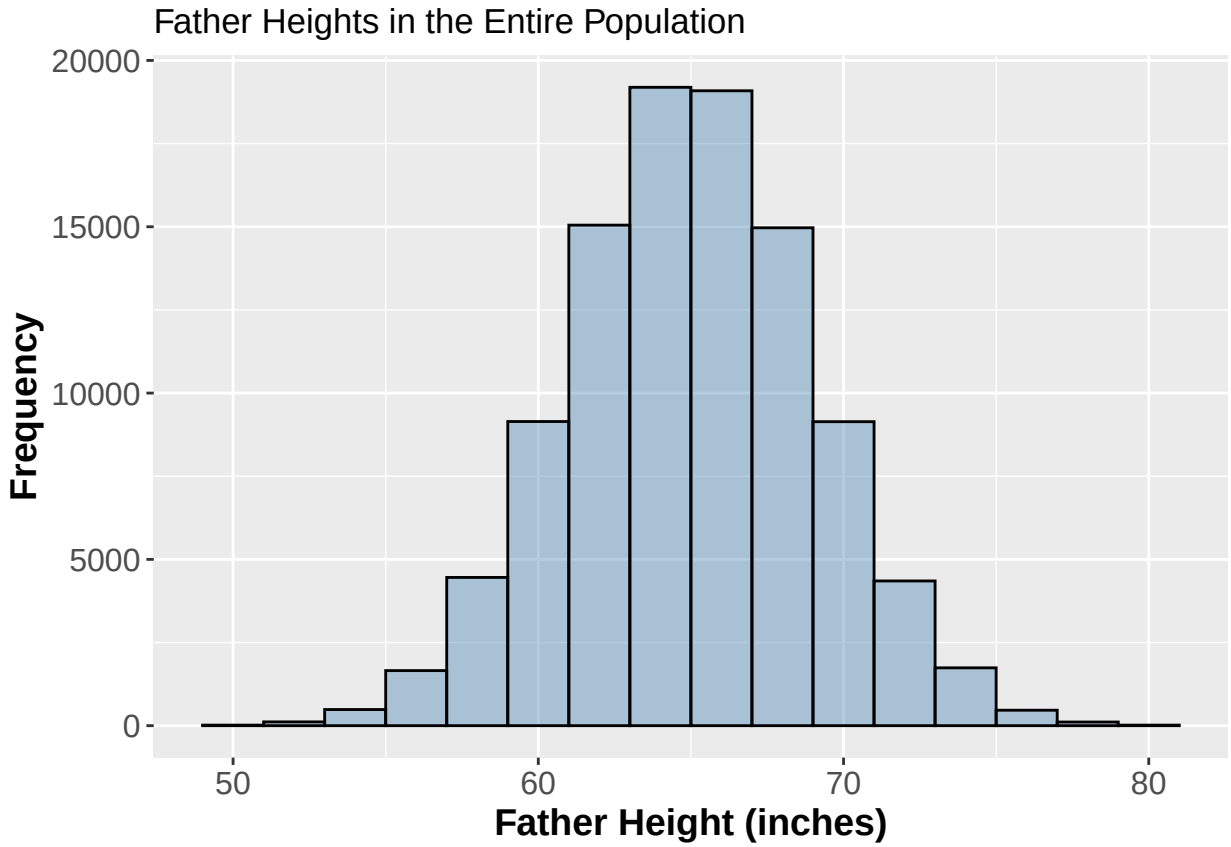



```
# Sample from a population of cars from the fuel efficiency mtcars dataset
nsamples = 32
carDataSample = carDataPopNoisy[sample(nrow(carDataPopNoisy), nsamples), ]
colnames(carDataSample) = c('hp', 'mpg')

ggplot(data = carDataSample, aes(x = hp, y = mpg)) +
   geom_point(size = 1, color = 'blue') +
   labs(x = 'HP', y = 'MPG') +
   #ggtitle("Fuel Efficiency vs. Horse Power for Random Samples") +
   theme(axis.text = element_text(size = 12),
   axis.text.x = element_text(size = 12),
   axis.text.y = element_text(size = 12),
   axis.title = element_text(size = 14, face = "bold"))
```



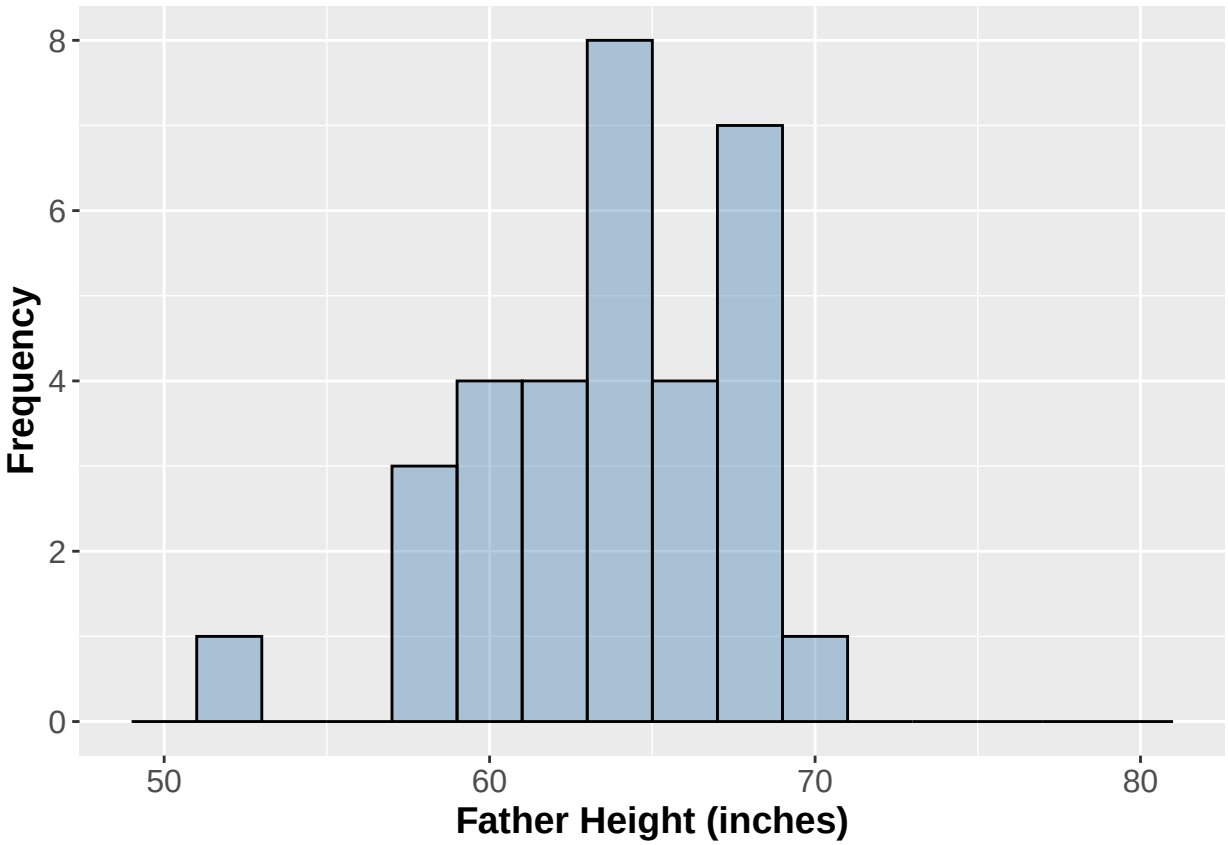

```
# Population data
mu_father = 65 # mean of father heights
sigma_father = 4 # standard deviation of father heights
popsize = 1e5 # population size
fatherHeights = rnorm(popsize, mean = mu_father, sd = sigma_father)

dfHeights = as.data.frame(fatherHeights)
colnames(dfHeights) = c('FatherHeight')

# Plot the frequency histogram for father heights
delta = 2.0 # bin width for histogram
ggplot(data = dfHeights) +
  geom_histogram(aes(x = FatherHeight, y = ..count..),
  breaks = seq(mu_father-4*sigma_father, mu_father+4*sigma_father, by = delta),
  colour = 'black', fill = 'steelblue', alpha = 0.4) +
  labs(x = 'Father Height (inches)', y = 'Frequency')+
  ggtitle('Father Heights in the Entire Population') +
   theme(axis.text = element_text(size = 12),
   axis.text.x = element_text(size = 12),
   axis.text.y = element_text(size = 12),
   axis.title = element_text(size = 14, face = "bold"))
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# Population mean of father's heights


# Draw random samples from the population
```

```
# Sample from a population of father heights
n = 32
fatherHeightsSample = dfHeights[sample(nrow(dfHeights), n), 'FatherHeight']
dfHeightsSample = as.data.frame(fatherHeightsSample)
colnames(dfHeightsSample) = c('FatherHeight')

# Plot the frequency histogram for father heights
delta = 2.0 # bin width for histogram
ggplot(data = dfHeightsSample) +
  geom_histogram(aes(x = FatherHeight, y = ..count..),
  breaks = seq(mu_father-4*sigma_father, mu_father+4*sigma_father, by = delta),
  colour = 'black', fill = 'steelblue', alpha = 0.4) +
  labs(x = 'Father Height (inches)', y = 'Frequency')+
  #ggtitle('Father Heights in a Sample of Size 32 Drawn from the Population') +
   theme(axis.text = element_text(size = 12),
   axis.text.x = element_text(size = 12),
   axis.text.y = element_text(size = 12),
   axis.title = element_text(size = 14, face = "bold"))
```


Frequency
8
6
4
2
0
50
60
70
80
Father Height (inches)