Exercise 1: We have aerial counts of bird nests from all $N = 1000$ plots that we have divided a region into. The total aerial count is $\tau_x = 4500$. We select a SRS of size $n = 10$ plots and visit them, performing a very careful 'ground truth' survey.

The results are:

```
plot_number <- 1:10
aerial_count <- c(5,0,0,3,3,2,8,6,4,4)
ground_count <- c(8,0,2,6,5,4,20,9,6,8)
df <- data.frame(plot_number, aerial_count, ground_count)
```

a. Find a 95 percent confidence interval for R, the ratio of ground count to aerial count (this is the reciprocal of the detection probability).

Solution:
Recall that the estimator for the ratio of random variables R is given by,

$$\hat{R} = \frac{\bar{y}}{\bar{x}}.$$

With a variance of,

$$\hat{V}(\hat{R}) = \frac{N-n}{N}\frac{1}{\mu_x^2}\frac{MSE_R}{n}.$$

Computing the 95 percent confidence interval get,
Code:

```
> R_estimator = mean(df$ground_count)/mean(df$aerial_count)
    [1] 1.942857
> MSE <- sum((df$ground_count - R_estimator*df$aerial_count)^2)
            /(length(df$ground_count) - 1)
    [1] 4.198277
> mu_x = 4500/1000
    [1] 4.5
> Variance_R = ((1000 - 10)/1000)*(1/mu_x^2)*(MSE/10)
    [1] 0.02052491
> ConfidenceInterval <- c(R_estimator + 2*sqrt(Variance_R),
                          R_estimator - 2*sqrt(Variance_R))
    [1] 2.229387 1.656327
```

b. Find a 95 percent confidence interval for τ_y, the total number of nests in the region.

Solution:
Recall that we can estimate τ_y by the following formula,

$$\hat{\tau}_y = \tau_x \frac{\bar{y}}{\bar{x}} = \tau_x \hat{R}$$

The variance can also be estimated with the following,

$$\hat{V}(\hat{\tau}_y) = N^2 \frac{N-n}{N} \frac{MSE_R}{n}.$$

Computing the 95 percent confidence interval for $\hat{\tau}_y$ we get,
Code:

```
> Tau_y = 4500*R_estimator
    [1] 8742.857
> Variance_Tau_y = (1000^2)*((1000 - 10)/1000)*(MSE/10)
    [1] 415629.4
> ConfidenceInterval <- c(Tau_y + 2*sqrt(Variance_Tau_y),
                          Tau_y - 2*sqrt(Variance_Tau_y))
    [1] 10032.244  7453.471
```

c. BONUS: Find a 95 percent confidence interval for $1/\tau_y$.

Exercise 2.: If we want to estimate the number of grass blades in a plot, and the plot varies a lot in grass density, we might try a rank set sample. Suppose we can't stand to count more than 4 quadrats (of 1 square inch each), and we think that we should be able to easily rank two samples just by looking at them. Describe how I would conduct a rank set sample in this situation (hint: we'll end up looking at 8 quadrats).

In an earlier assignment (Problem 5 in HW 3), you had a number of plots, where you had a guess at the grass cover at all locations (column 'Guess at Cover'), basically a census for a simple-to-assign variable. In that problem we used the guesses to stratify the population. We could, instead, just collect a SRS and use ratio estimation to get an improved estimate of cover everywhere.

You know the guess at grass cover for all the plots:

```
guess <- c(0.4, 0.3, 0.5, 0.7, 0.8, 0.1, 0.3,
 0.6, 0.5, 0.3, 0.8, 0.9, 0.6, 0.6, 0.5, 0.6,
 0.7, 0.2, 0.5, 0.8, 0.2, 0.7, 0.6, 1, 0.7,
 0.2, 0.7, 0, 0.5, 0.1, 0.2, 0.8, 0.7, 0.2,
 0.1, 0.4, 0.8, 0.5, 0.2, 0.6)
```

a. Take a simple random sample of size $n = 8$ from the population of plots. In real life you would then go to those plots and measure the true grass cover Y. In this assignment you should instead go to the earlier problem (PROBLEM 5 in HW 3) and get the Y values. Then obtain a 95 percent confidence interval for the **mean** grass cover using the ratio estimator approach.

Solution:
Going back to the data from homework 3, we can conduct a SRS $n = 8$ on the true grass cover, and obtaining a mean grass cover estimator using the ratio estimator.
Code:

```
> df <- read.csv('tabula-HW3stat402.csv')
> SRS <- df[sample(nrow(df), 8),]
    Plot True.Cover Guess.at.Cover
1      1       0.40            0.4
7      7       0.15            0.3
39    39       0.25            0.2
32    32       0.85            0.8
33    33       0.70            0.7
17    17       0.55            0.7
26    26       0.10            0.2
35    35       0.10            0.1
> est.R = mean(SRS$True.Cover)/mean(SRS$Guess.at.Cover)
     [1] 0.9117647
> mu_x = mean(df$Guess.at.Cover)
     [1] 0.4975
> est.mu_y = est.R*mu_x
     [1] 0.4536029

> MSE <- sum((SRS$True.Cover - est.R*SRS$Guess.at.Cover)^2)
              /(length(SRS$Guess.at.Cover) - 1)
     [1] 0.007726149

> est.V = ((length(df$Guess.at.Cover) - 8)/
              length(df$Guess.at.Cover))
              *(MSE/8)
     [1] 0.0007726149
> ConfidenceInterval <- c(est.mu_y + 2*sqrt(est.V),
```

```
                                    est.mu_y - 2*sqrt(est.V))
    [1] 0.5091948 0.3980110
```

b. In a tragic accident, your research assistant lost all of the 'guess' measurements. Repeat the analysis using only the 'true cover' values from your sample and the usual 95 percent confidence interval for the mean based on a SRS.
Code:

```
> mu_x = mean(SRS$Guess.at.Cover)
    [1] 0.425
> est.mu_y = est.R*mu_x
    [1] 0.3875
> ConfidenceInterval <- c(est.mu_y + 2*sqrt(est.V),
                          est.mu_y - 2*sqrt(est.V))
    [1] 0.4430919 0.3319081
```

c. The true average cover is 0.46. Did both of your intervals contain the true average? How do the two intervals compare in width?

Solution:
The second analysis, which estimated the mean guess cover ended up not containing the true average cover. The confidence interval was shifted down when we used the estimated mean guess cover so it must be that our samples skewed lower than the true mean.

Exercise 3.: Using the sample from problem 2, compute a 95 percent confidence interval for the mean using the regression estimator. How does it compare to the other approaches?

Solution:
Code:

```
> SRS_Regression <- lm(True.Cover ~ Guess.at.Cover, data = SRS)

lm(formula = True.Cover ~ Guess.at.Cover, data = SRS)
```

```
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     -0.04369    0.06210  -0.704 0.508104
Guess.at.Cover   1.01456    0.12546   8.086 0.000192 ***

Residual standard error: 0.09004 on 6 degrees of freedom
Multiple R-squared:  0.916,     Adjusted R-squared:  0.9019
F-statistic: 65.39 on 1 and 6 DF,  p-value: 0.0001917

> est.mu_y = mean(SRS$True.Cover) + 1.01456*
             (mean(df$Guess.at.Cover) - mean(SRS$Guess.at.Cover))
    [1] 0.4610556

> anova(SRS_Regression)
Analysis of Variance Table
Response: True.Cover
                Df  Sum Sq Mean Sq F value    Pr(>F)
Guess.at.Cover   1 0.53011 0.53011  65.391 0.0001917 ***
Residuals        6 0.04864 0.00811

> MSE = 0.00811
> est.V = ((length(df$Guess.at.Cover) - 8)/
           length(df$Guess.at.Cover))*(MSE/8)
    [1] 0.000811

> ConfidenceInterval <- c(est.mu_y + 2*sqrt(est.V),
                          est.mu_y - 2*sqrt(est.V))
    [1] 0.5180117 0.4040995
```

The regression estimator introduces a little more variance than either ratio estimator. I do think that the regression estimator is slightly more centered around the true mean, that we know from the data to be around .46 which is almost exactly our estimate of the mean.

Exercise 4.: Sometimes you can transform data to get a better estimate. Suppose there are $N = 200$ plots, and you take an inexpensive estimate of contaminate level in all plots (the is X). You also can sample $n = 30$ plots and take a highly accurate measurement Y. Unfortunately, the relationship is not linear. Here is the pertinent information:
All X values in parts per thousand (inexpensive estimate):

```
X <- c(4.2, 4.9, 0.5, 4.3, 6, 0.5, 6.7, 1.1, 5.6,
3.1, 3.6, 8.1, 3.7, 3.8, 7.3, 4, 0.5, 5.9, 9.3, 8,
2, 1.4, 3.6, 1.9, 1.2, 4.3, 1.6, 0.5, 5.8, 4.7, 3.7,
8.3, 4.9, 7.8, 5.5, 0.5, 3.9, 7.2, 6.5, 3.5, 7, 3,
0.5, 3.7, 0.5, 7.5, 4.6, 1.9, 4.3, 8.2, 5.2, 3.8, 4.3,
0.5, 2.6, 2.5, 3.3, 4, 4, 2.4, 2.7, 9.3, 7, 0.5, 3.6,
6.8, 0.5, 0.5, 2.6, 9.6, 4, 0.5, 6.7, 6.3, 5.2, 0.5,
6.2, 3.8, 6.3, 4.9, 5, 1.9, 3.4, 1.3, 3.7, 3, 5.5, 4.4,
3.1, 6, 2.5, 5.4, 5.3, 3.9, 2.6, 4.9, 2.9, 4.5, 6.8, 0.5,
3.7, 6.3, 7.8, 1.9, 0.5, 4.8, 3.2, 0.5, 6.8, 5, 2.1, 3.9,
2.4, 10.8, 6.1, 6, 9.9, 1.5, 2, 3.3, 5.8, 2.9, 6.4, 4.4,
1.9, 10.6, 3, 4.8, 4.2, 4.3, 3.1, 2.9, 3.5, 2.4, 3.3, 3.8,
5.8, 3.5, 0.5, 5, 3.9, 4.9, 1.7, 4.5, 0.5, 6.5, 0.5, 1.5,
1.9, 1.9, 3.8, 0.5, 3.6, 2.8, 0.5, 4.2, 2.4, 2.9, 3.9, 0.5,
2.7, 10.5, 2.7, 7.9, 0.5, 3.3, 4.8, 3.5, 2.1, 6.8, 7.2, 5.1,
2, 4.6, 3.6, 5.2, 6, 6.3, 7.1, 2.5, 3.2, 1.4, 3.8, 1.4, 4.6,
7.6, 2.7, 9.6, 2.2, 3.4, 2, 3.3, 2.9, 4.6, 5.4, 6.7, 1.9, 0.5,
2.8, 4.9)
```

I drew a SRS from this population and also collected the corresponding Y value:

```
x_samp <- c(4.2, 6.4, 7.9, 2.9, 2.9, 1.9, 7.8, 6,
1.6, 2.4, 4, 10.8, 1.7, 1.2, 2.1, 0.5, 4.6, 0.5, 7,
0.5, 3.5, 1.1, 6.2, 3.6, 3.1, 5.8, 2.4, 3.3, 4.9, 4.9)

y_samp <- c(12, 38, 34, 4, 8, 3, 29, 43, 1, 5,
14, 71, 1, 1, 6, 0, 18, 0, 41, 0, 9, 1, 14, 13,
8, 21, 2, 8, 25, 27)
```

a. Plot x_samp vs y_samp. It is not really linear. Can you find a transformation of X that makes the plot more linear? Apply it and then plot the result.

Solution:
Plotting the data we get the following, and transforming the x_samp data by squaring it we actually get a higher correlation. The original data gets a correlation of 0.9260864 and the transformed data get 0.9307337.

Figure 1: Plotted Data

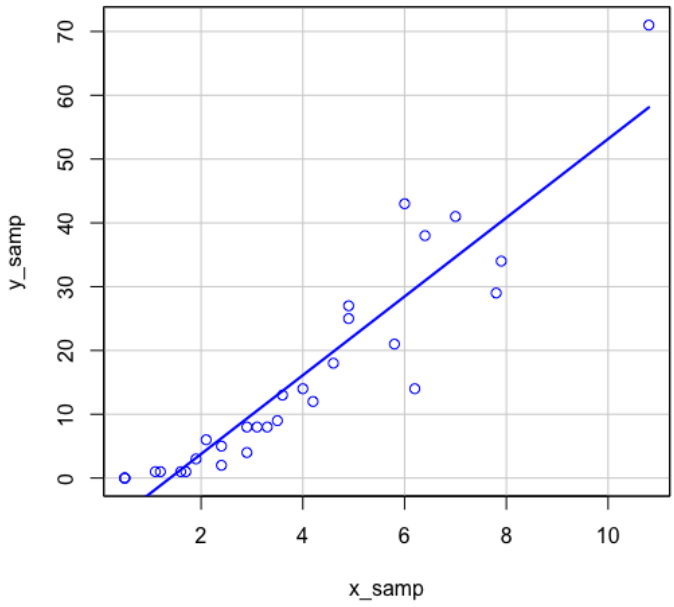
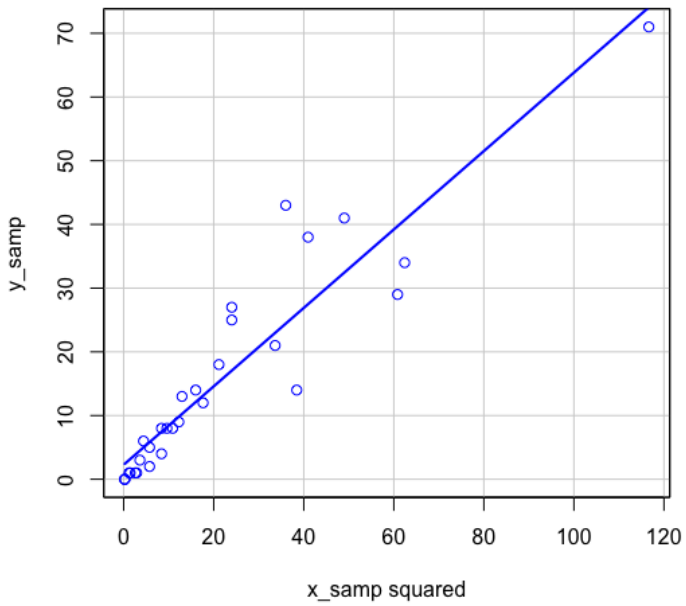


Figure 2: Transformed Data(Squaring X)



Code:

```
> cor(x_samp, y_samp)
    [1] 0.9260864
> cor(x_samp^2, y_samp)
    [1] 0.9307337

> scatterplot(x_samp^2, y_samp,
+             regLine = TRUE, boxplots = FALSE,
+             smooth = FALSE,
             xlab = 'x_samp squared', ylab = 'y_samp')
> scatterplot(x_samp, y_samp,
+             regLine = TRUE, boxplots = FALSE,
+             smooth = FALSE, xlab = 'x_samp', ylab = 'y_samp')
```

b. Apply this transformation to all of the X values to find the new (transformed) τ_{x*}. Now use either the ratio or regression estimator (your choice) to get a 95 percent confidence interval for the average contaminant level in the area.

Solution:
Applying the ratio estimator on the transformed data, we get **Code:**

```
> est.R = mean(y_samp)/mean((x_samp)^2)
    [1] 0.723479

> mu_x = mean(X^2)
    [1] 21.622

> est.mu_y = est.R*mu_x
    [1] 15.64306

> MSE <- sum((y_samp - est.R*(x_samp^2))^2)
            /(length(y_samp) - 1)
    [1] 45.57133

> est.V = ((length(X) - length(y_samp))/length(X))
            *(MSE/length(y_samp))
    [1] 1.291188

> ConfidenceInterval <- c(est.mu_y + 2*sqrt(est.V),
                          est.mu_y - 2*sqrt(est.V))
    [1] 17.91567 13.37045
```

Applying a linear model to the transformed data we get an intercept of around -8 with very high significance. That tells me that we will likely get a more accurate estimator for the mean containment level. Doing so we get,
Code:

```
> SRS_Regression <- lm(y_samp ~ x_samp^2)
Call:
lm(formula = y_samp ~ x_samp^2)

Residuals:
     Min       1Q   Median       3Q      Max
-15.7000  -3.9728  -0.7839   4.8256  14.5347

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -8.5759     2.1812  -3.932 0.000504 ***
x_samp        6.1735     0.4753  12.988 2.25e-13 ***
---

Residual standard error: 6.473 on 28 degrees of freedom
Multiple R-squared:  0.8576,    Adjusted R-squared:  0.8526
F-statistic: 168.7 on 1 and 28 DF,  p-value: 2.255e-13

> est.mu_y = mean(y_samp) + 6.1735*(mean(X^2) - mean(x_samp^2))
    [1] 18.72959


> anova(SRS_Regression)
Analysis of Variance Table

Response: y_samp
           Df Sum Sq Mean Sq F value    Pr(>F)
x_samp      1 7068.1  7068.1  168.68 2.255e-13 ***
Residuals 28 1173.3    41.9
---
> MSE = 41.9
> est.V = ((length(X) - length(y_samp))/length(X))*
            (MSE/length(y_samp))
    [1] 1.187167

> ConfidenceInterval <- c(est.mu_y + 2*sqrt(est.V),
                          est.mu_y - 2*sqrt(est.V))
    [1] 20.90874 16.55045
```