Lab 4

Yunting Chiu

2021-02-05

R Lab 4

- Read the build-in cars dataframe

```
library(tidyverse)
```

```
## -- Attaching packages --- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

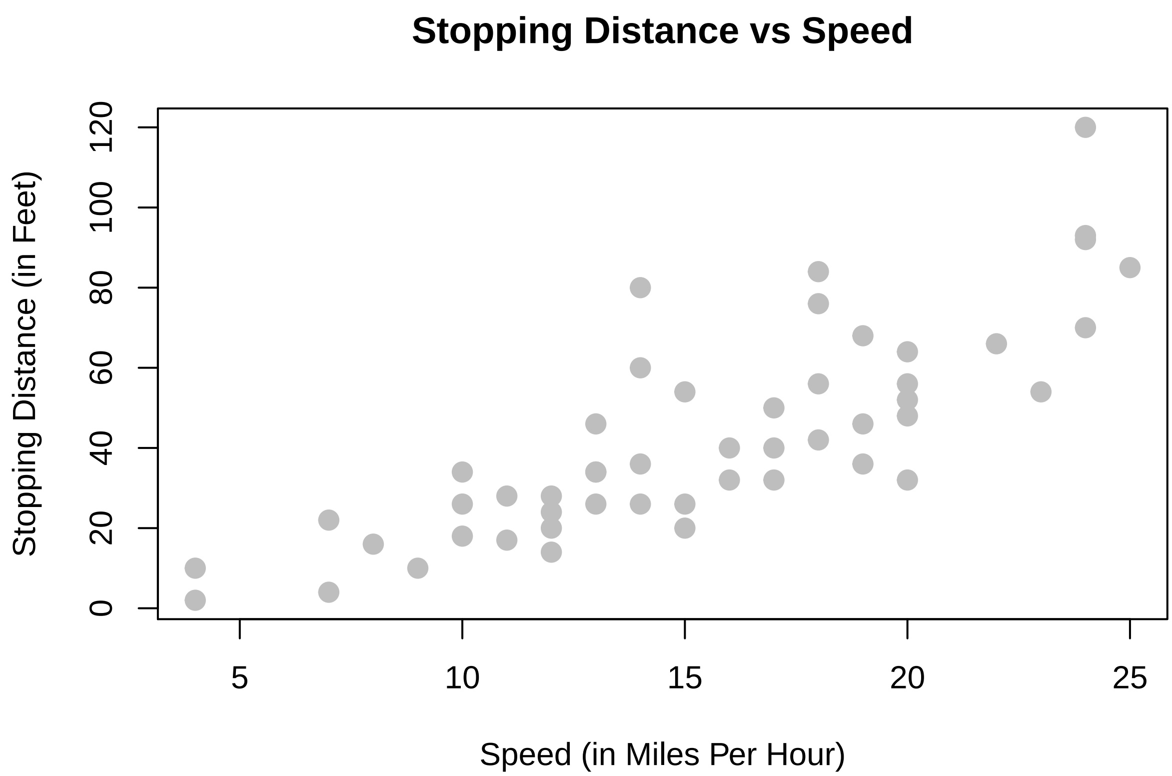
```
cars #dist = distance
```

```
##    speed dist
## 1      4    2
## 2      4   10
## 3      7    4
## 4      7   22
## 5      8   16
## 6      9   10
## 7     10   18
## 8     10   26
## 9     10   34
## 10    11   17
## 11    11   28
## 12    12   14
## 13    12   20
## 14    12   24
## 15    12   28
## 16    13   26
## 17    13   34
## 18    13   34
## 19    13   46
## 20    14   26
## 21    14   36
## 22    14   60
## 23    14   80
## 24    15   20
## 25    15   26
```

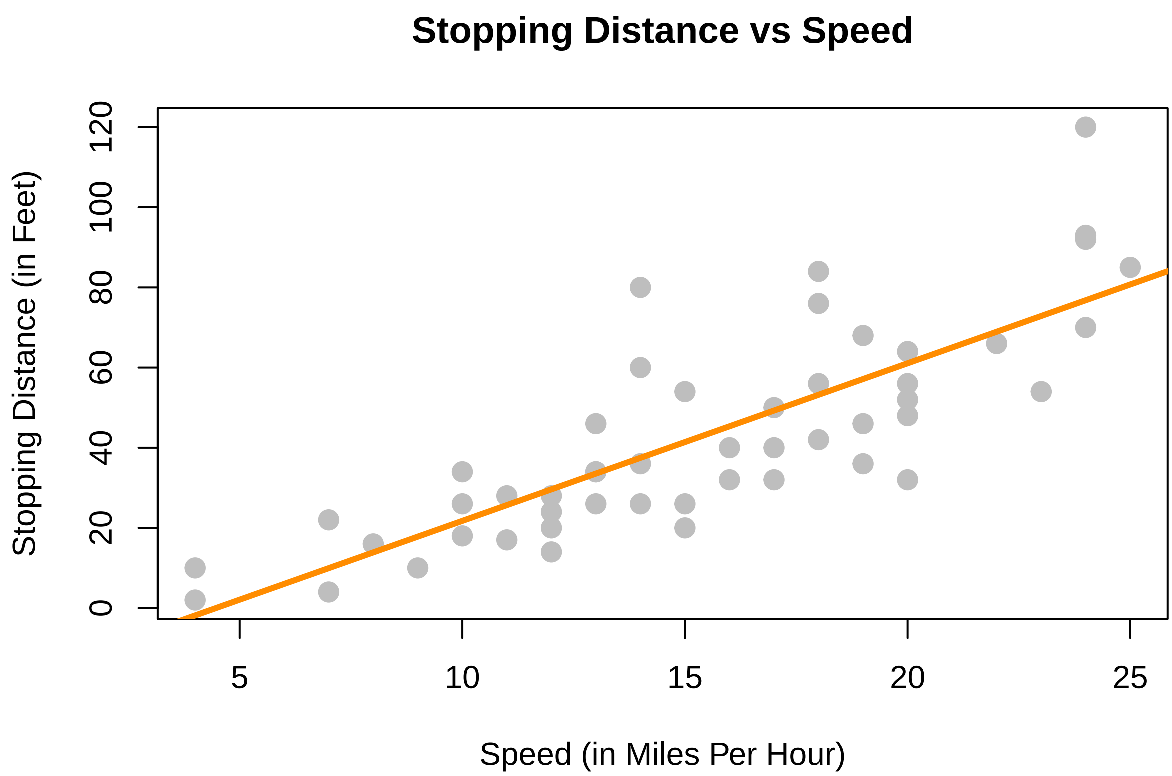
```
## 26    15   54
## 27    16   32
## 28    16   40
## 29    17   32
## 30    17   40
## 31    17   50
## 32    18   42
## 33    18   56
## 34    18   76
## 35    18   84
## 36    19   36
## 37    19   46
## 38    19   68
## 39    20   32
## 40    20   48
## 41    20   52
## 42    20   56
## 43    20   64
## 44    22   66
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
```

- Plot the two variables, independent variable is Speed (in Miles Per Hour) and dependent variable is Stopping Distance (in Feet).

```r
plot(dist ~ speed, data = cars,
xlab = "Speed (in Miles Per Hour)",
ylab = "Stopping Distance (in Feet)",
main = "Stopping Distance vs Speed",
pch = 20,
cex =2,
col = "grey")
```




```
stop_dist_model = lm(dist ~ speed, data = cars)
plot(dist ~ speed, data = cars,
     xlab = "Speed (in Miles Per Hour)",
     ylab = "Stopping Distance (in Feet)",
     main = "Stopping Distance vs Speed",
     pch  = 20,
    cex =2,
    col = "grey")
abline(stop_dist_model, lwd = 3, col = "darkorange")
```



Fit the model

```
stop_dist_model = lm(dist ~ speed, data = cars)
summary(stop_dist_model)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Tests in R

- find the `coefficients` table so we use `names` function to see the all tables in summary.

```r
names(summary(stop_dist_model))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

- We can find b0 = -17.579095 and b1 = 3.9324088 from the coefficients table below.
- b0 and b1 are the estimators for the model by $\beta 0$ and $\beta 1$.

```r
summary(stop_dist_model)$coefficients
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) -17.579095  6.7584402 -2.601058 1.231882e-02
## speed         3.932409  0.4155128  9.463990 1.489836e-12
```

$\beta 1$

- Let we focus on $\beta 1$ first so we need to extract $\beta 1$ row.

```r
summary(stop_dist_model)$coefficients[2, ] #[row, column]
```

```
##     Estimate   Std. Error      t value     Pr(>|t|)
## 3.932409e+00 4.155128e-01 9.463990e+00 1.489836e-12
```

- Estimate b1 is 3.932409e+00
- Standard error of b1 is 0.4155128
- t-value, which is testing for null hypothesis.
- p-value = 1.489836e-12

$\beta 0$

- Extract $\beta 0$ row from coefficients table

```r
summary(stop_dist_model)$coefficients[1,]
```

```
##     Estimate   Std. Error      t value     Pr(>|t|)
## -17.57909489   6.75844017  -2.60105800   0.01231882
```

create new factors

```r
stop_dist_model_test_info = summary(stop_dist_model)$coefficients
b_0 = stop_dist_model_test_info[1, 1] # Estimate
b_0_se = stop_dist_model_test_info[1, 2] # Std. Error
b_0_t = stop_dist_model_test_info[1, 3] # t value
b_0_pval = stop_dist_model_test_info[1, 4] # Pr(>|t|)

b_1 = stop_dist_model_test_info[2, 1] # Estimate
b_1_se = stop_dist_model_test_info[2, 2] # Std. Error
b_1_t = stop_dist_model_test_info[2, 3] # t value
b_1_pval = stop_dist_model_test_info[2, 4] # Pr(>|t|)
```

Manually Task

- t-statistic for b1 by hand

```r
(b_1 - 0) / b_1_se
```

```
## [1] 9.46399
```

- From coefficients table

```r
b_1_t
```

```
## [1] 9.46399
```

- p-value by hand

```r
2 * pt(abs(b_1_t), df = length(resid(stop_dist_model)) - 2, lower.tail = FALSE)
```

```
## [1] 1.489836e-12
```

- From coefficients table

```r
b_1_pval
```

```
## [1] 1.489836e-12
```

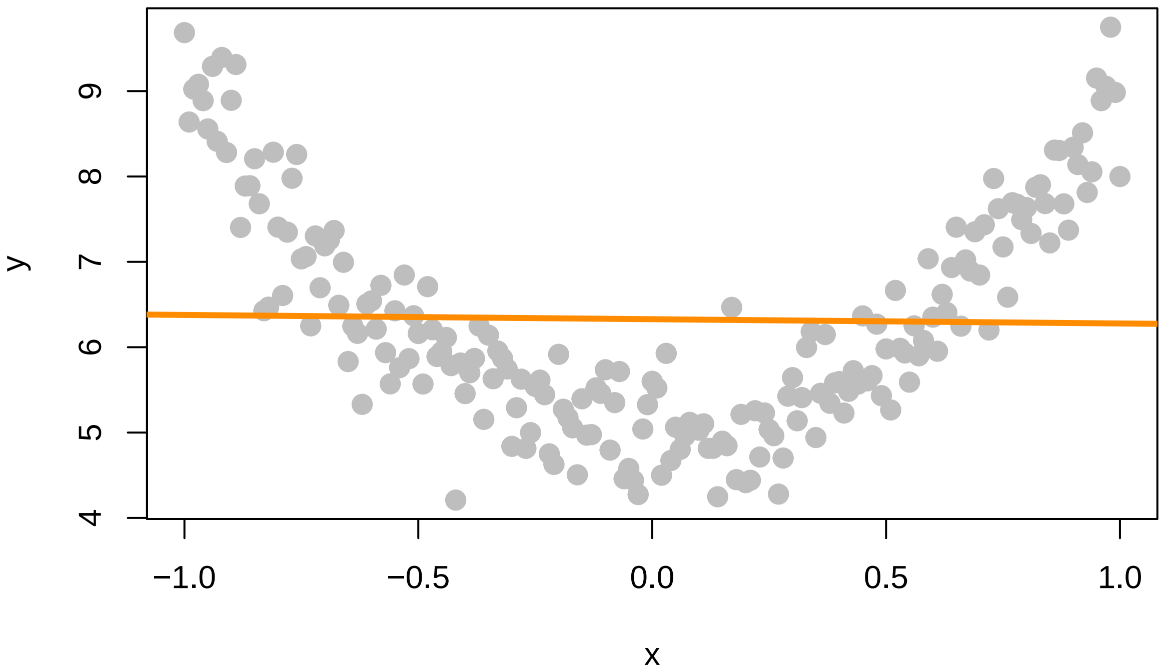
Significance of Regression, t-Test

- For the `cars` example:
  - Under H0 there is not a significant linear relationship between speed and stopping distance.
  - Under H1 there is a significant linear relationship between speed and stopping distance.

That is, we need to know the expected value of b1 is an unbiased estimator for $\beta 1$. We set H0 : $\beta 1 = 0$ vs H1 : $\beta 1$!= 0. With the small p-value, we have evidence to reject the null in favor of alternative hypothesis. Thus, the $\beta 1$ is not equal to zero so there is a significant linear relation between `Speed (in Miles Per Hour)` and `Stopping Distance (in Feet)`.

Know what is the linear

```r
set.seed(42)
x = seq(-1, 1, 0.01)
y = 5 + 4 * x ^ 2 + rnorm(length(x), 0, 0.5)
plot(x, y, pch = 20, cex = 2, col = "grey")
abline(lm(y ~ x), lwd = 3, col = "darkorange")
```



- Let we run the linear model and set: H0 : $\beta 1 = 0$ vs H1 : $\beta 1 != 0$ - With the large p-value 0.756 explained below, we fail to reject the null. In other words, there is no significant linear relationship between x and y.

```r
regSec <- lm(y~x)
regSec
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##     6.32802     -0.05006
```

```r
summary(regSec)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1400 -1.0015 -0.3147  0.9806  3.4703
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.32802    0.09352  67.666   <2e-16 ***
## x           -0.05006    0.16118  -0.311    0.756
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.326 on 199 degrees of freedom
## Multiple R-squared:  0.0004844,  Adjusted R-squared:  -0.004538
## F-statistic: 0.09645 on 1 and 199 DF,  p-value: 0.7565
```

Confidence Intervals in R

Using `confint` function then we can smoothly get the confidence intervals for $\beta0$ and $\beta1$.

```
# 99% cinfidence interval
confint(stop_dist_model, level = 0.99)
```

```
##                   0.5 %    99.5 %
## (Intercept) -35.706610 0.5484205
## speed         2.817919 5.0468988
```

- In this case, if the car increase in 1 mile/per hour, the Stopping Distance (in Feet) will increase in 2.817919 to 5.0468988 feet with 99 % confidence interval. We have 99% confident to explain it.

Extract the results separately

```
confint(stop_dist_model, level = 0.99)[1,]
```

```
##       0.5 %      99.5 %
## -35.7066103   0.5484205
```

```
confint(stop_dist_model, level = 0.99)[1, 1]
```

```
## [1] -35.70661
```

```
confint(stop_dist_model, level = 0.99)[1, 2]
```

```
## [1] 0.5484205
```

```
confint(stop_dist_model, parm = "(Intercept)", level = 0.99)
```

```
##                 0.5 %    99.5 %
## (Intercept) -35.70661 0.5484205
```

```
confint(stop_dist_model, level = 0.99)[2,]
```

```
##    0.5 %   99.5 %
## 2.817919 5.046899
```

```
confint(stop_dist_model, level = 0.99)[2, 1]
```

```
## [1] 2.817919
```

```
confint(stop_dist_model, level = 0.99)[2, 2]
```

```
## [1] 5.046899
```

```
confint(stop_dist_model, parm = "speed", level = 0.99)
```

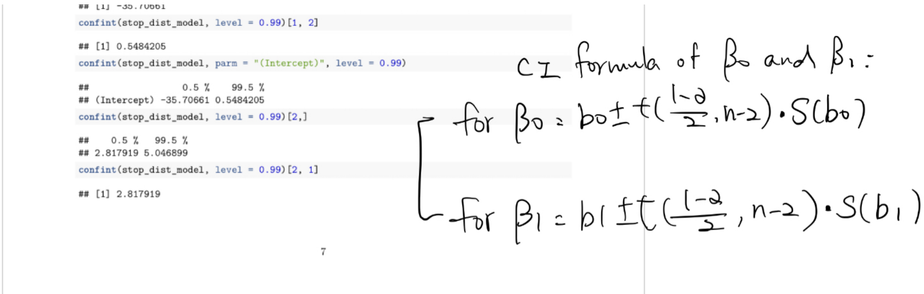
```
##          0.5 %   99.5 %
## speed 2.817919 5.046899
```

Verify that calculations that R is performing for the $\beta1$ interval.

```
# store estimate
b_1 = coef(stop_dist_model)[2] # store standard error
b_1_se = summary(stop_dist_model)$coefficients[2, 2] # calculate critical value for two-sided 99% CI
crit = qt(0.995, df = length(resid(stop_dist_model)) - 2) # est - margin, est + margin
c(b_1 - crit * b_1_se, b_1 + crit * b_1_se)
```

```
##    speed    speed
## 2.817919 5.046899
```

Note



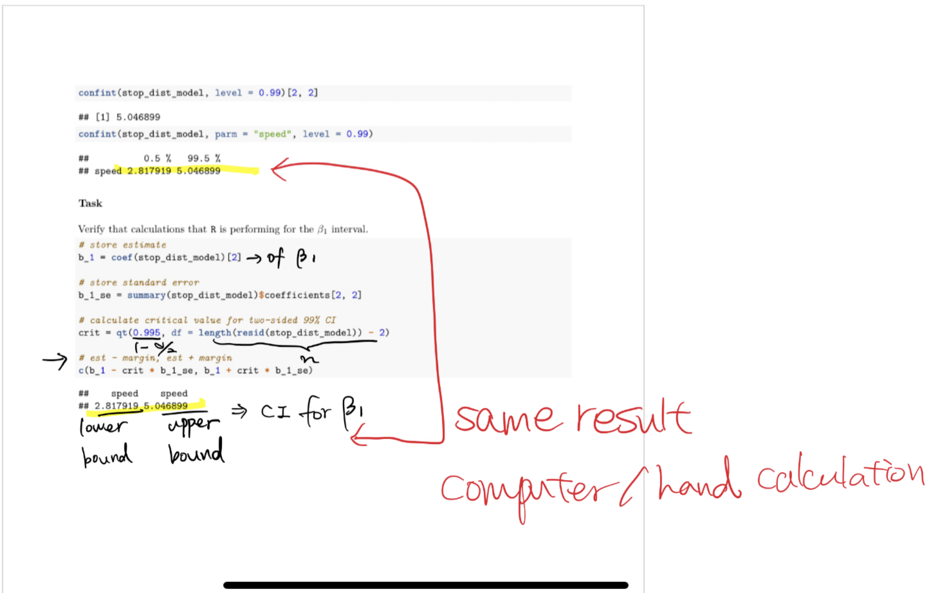
```
## [1] -35.70661
confint(stop_dist_model, level = 0.99)[1, 2]

## [1] 0.5484205
confint(stop_dist_model, parm = "(Intercept)", level = 0.99)

##                  0.5 %    99.5 %
## (Intercept) -35.70661 0.5484205
confint(stop_dist_model, level = 0.99)[2,]

##    0.5 %   99.5 %
## 2.817919 5.046899
confint(stop_dist_model, level = 0.99)[2, 1]

## [1] 2.817919
```

CI formula of β_0 and β_1:

for $\beta_0 = b_0 \pm t(\frac{1-\alpha}{2}, n-2) \cdot S(b_0)$

for $\beta_1 = b_1 \pm t(\frac{1-\alpha}{2}, n-2) \cdot S(b_1)$

7



```
confint(stop_dist_model, level = 0.99)[2, 2]

## [1] 5.046899
confint(stop_dist_model, parm = "speed", level = 0.99)

##          0.5 %   99.5 %
## speed 2.817919 5.046899
```

Task

Verify that calculations that R is performing for the β_1 interval.

```
# store estimate
b_1 = coef(stop_dist_model)[2]  → of β1

# store standard error
b_1_se = summary(stop_dist_model)$coefficients[2, 2]

# calculate critical value for two-sided 99% CI
crit = qt(0.995, df = length(resid(stop_dist_model)) - 2)
           1-α/2            n

# est - margin, est + margin
c(b_1 - crit * b_1_se, b_1 + crit * b_1_se)

##    speed    speed
## 2.817919 5.046899   ⇒ CI for β1
   lower     upper
   bound     bound
```

same result

Computer / hand calculation