# Rudimentary R Skills

Tsering Sherpa

2025-06-06

**Basic R Skills: Examples using built in dataset "iris"** #Quick view

```
data("iris")
#Name the data
iris_df<-iris
#view first 6 rows
head(iris_df)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
#view last 6 rows
tail(iris_df)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

#Descriptive stats

```
#Compute mean using r function
mean(iris_df$Petal.Length)
```

```
## [1] 3.758
```

```
#Compute standard deviation using r function

sd(iris_df$Petal.Length)
```

```
## [1] 1.765298
```

```r
#Compute standard deviation using r with formula for standard deviation (r can be used like a calculato

sqrt(sum( (iris_df$Petal.Length-mean(iris_df$Petal.Length))^2 )/( length(iris_df$Petal.Length)-
1) )
```

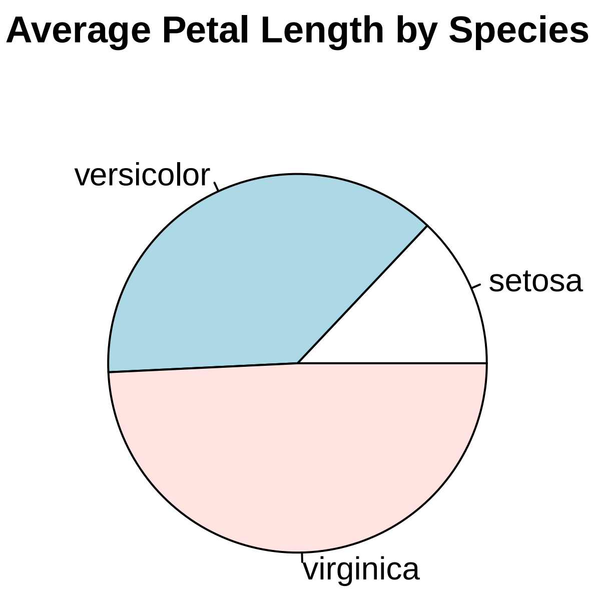
```
## [1] 1.765298
```

```r
#Quick overview of iris data
summary(iris_df)
```

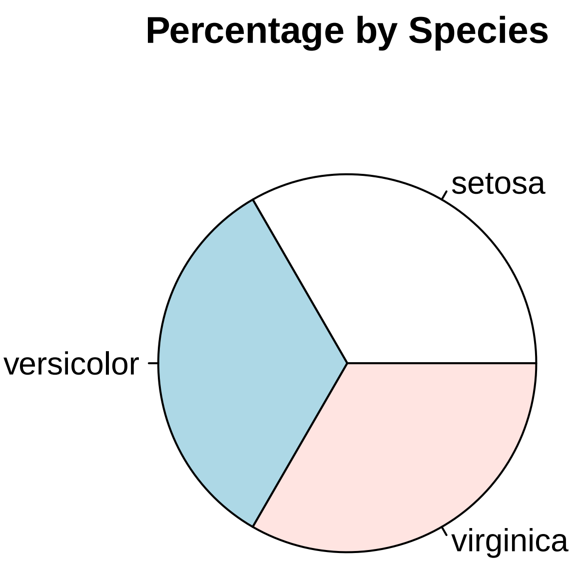
```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
## 
```

## Pie chart: Example using average petal length by species (not raw values)

```r
species <- c("setosa", "versicolor", "virginica")
avg_petal_length <- c(1.462, 4.260, 5.552)
pie(avg_petal_length, labels = species, main = "Average Petal Length by Species")
```

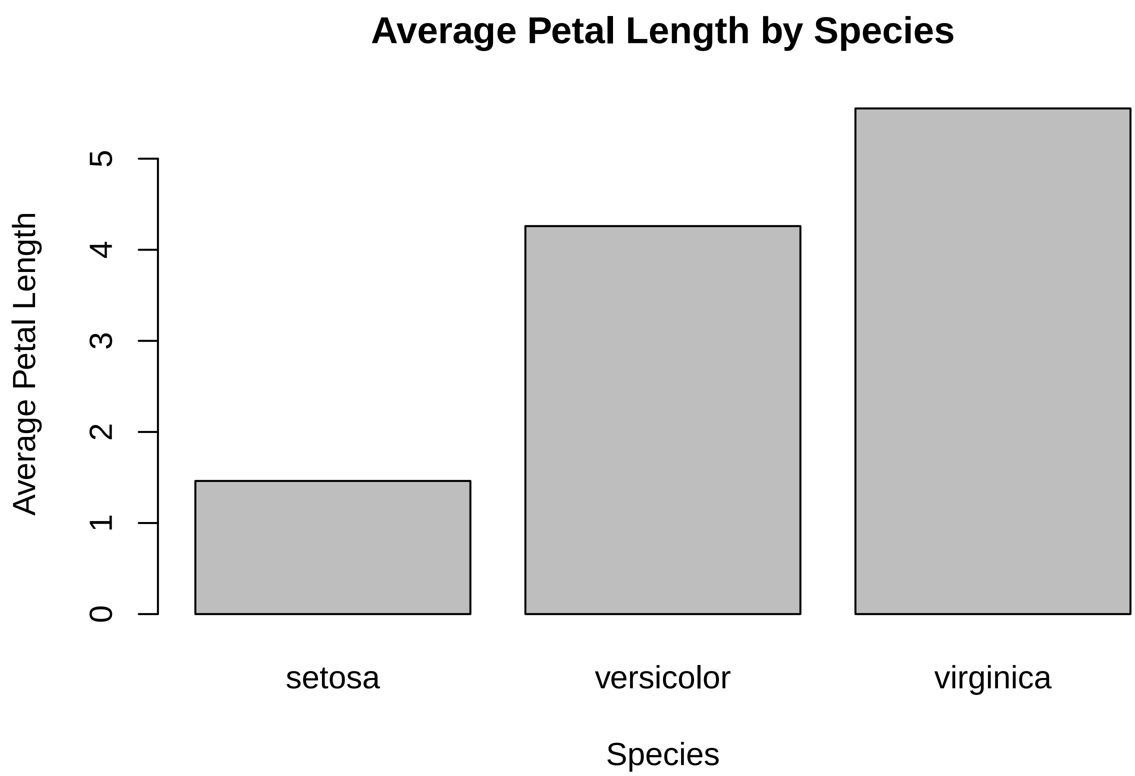


```r
# Pie chart: percentage for each species
species <- c("setosa", "versicolor", "virginica")
percent <- c(33, 33, 33)

pie(percent, labels = species, main = "Percentage by Species")
```



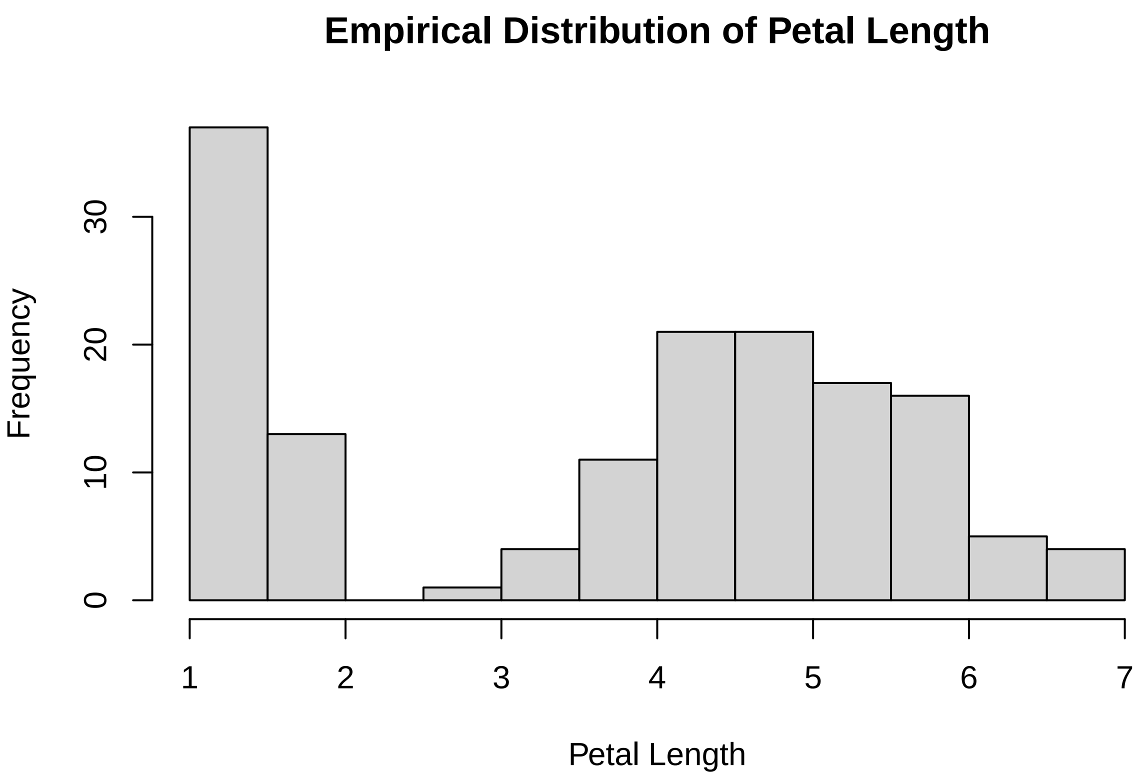

#Bar Charts in R: Iris petal length

```r
# Calculate average petal length by species
avg_petal <- tapply(iris$Petal.Length, iris$Species, mean)

# Create bar chart
barplot(avg_petal,
        names.arg = names(avg_petal),
        xlab = "Species",
        ylab = "Average Petal Length",
        main = "Average Petal Length by Species")
```




### Histogram of Petal Length from the iris dataset

```
hist(iris$Petal.Length,
     xlab = "Petal Length",
     ylab = "Frequency",
     main = "Empirical Distribution of Petal Length")
```




#Example: Contingency Table of Petal Length Category by Species

```
# Convert Petal.Length into categories
iris$PetalCat <- cut(iris$Petal.Length,
                     breaks = c(0, 2, 5, 7),
                     labels = c("Short", "Medium", "Long"),
                     right = FALSE)

# Create contingency table
petal_table <- table(iris$Species, iris$PetalCat)

# Add margins (totals)
addmargins(petal_table)
```

```
##
##               Short Medium Long Sum
##    setosa        50      0    0  50
##    versicolor     0     48    2  50
##    virginica      0      6   44  50
##    Sum           50     54   46 150
```

#Creating a Stem Plot

```
stem(iris_df$Petal.Length,scale = 1, width = 80, atom = 1e-08)
```

```
##
##    The decimal point is at the |
##
```

```
##   1 | 01223333333444444444444
##   1 | 555555555555556666666777799
##   2 |
##   2 |
##   3 | 033
##   3 | 55678999
##   4 | 000001112222334444
##   4 | 5555555566677777888899999
##   5 | 000011111111223344
##   5 | 5556666667778889 9
##   6 | 0011134
##   6 | 6779
```

## Confidence Intervals and T-test: Using built-in dataset "women"

```r
data("women")  # This includes height and weight for 15 women

#Stem plot of the weight variable
stem(women$weight)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   11 | 57
##   12 | 0369
##   13 | 259
##   14 | 26
##   15 | 049
##   16 | 4
```

```r
# Comment:
# A sample size of at least 15 allows us to use t-procedures if the data show no strong skewness or out
# This stem plot does not show strong skewness or clear outliers, so the t-procedure is appropriate eve
```

```r
# ----------------------------------------
# 2. Constructing a 95% Confidence Interval for the Population Mean of Weight
# ----------------------------------------

# Sample mean and standard deviation
mean_weight <- mean(women$weight)
sd_weight <- sd(women$weight)

# Sample size
n <- length(women$weight)

# Degrees of freedom
df <- n - 1

# Find the t* critical value
t_star <- qt(0.975, df)
```

```r
# Compute margin of error
margin_error <- t_star * (sd_weight / sqrt(n))

# Confidence interval
lower_bound <- mean_weight - margin_error
upper_bound <- mean_weight + margin_error

# Print results
mean_weight
```

```
## [1] 136.7333
```

```r
sd_weight
```

```
## [1] 15.49869
```

```r
t_star
```

```
## [1] 2.144787
```

```r
margin_error
```

```
## [1] 8.582891
```

```r
c(lower_bound, upper_bound)
```

```
## [1] 128.1504 145.3162
```

```r
# Interpretation:
# This gives us a 95% confidence interval for the population mean of weight:
# (128.1504, 145.3162)
```

#Hypothesis Test: H0: mean (population) = 62.5 vs Ha: mean (population) does not equal 62.5

```r
t.test(women$height, mu = 62.5, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  women$height
## t = 2.1651, df = 14, p-value = 0.04815
## alternative hypothesis: true mean is not equal to 62.5
## 95 percent confidence interval:
##  62.52341 67.47659
## sample estimates:
## mean of x
##        65
```

```r
# Interpretation:
# The p-value is 0.04815. If we use alpha = 0.05, this is just small enough to reject the null.
# That means there's moderate evidence that the true mean height differs from 62.5.
```

## Explore how the CI relates to different null hypothesis values

```r
# Recall our CI was: (62.52, 67.48)
# Let's try a null value that lies inside this interval

# Try mu = 66.5
t.test(women$height, mu = 66.5, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  women$height
## t = -1.299, df = 14, p-value = 0.2149
## alternative hypothesis: true mean is not equal to 66.5
## 95 percent confidence interval:
##  62.52341 67.47659
## sample estimates:
## mean of x
##        65
```

```r
# Try mu = 65 (the sample mean)
t.test(women$height, mu = 65, alternative = "two.sided")
```

```
##
##  One Sample t-test
##
## data:  women$height
## t = 0, df = 14, p-value = 1
## alternative hypothesis: true mean is not equal to 65
## 95 percent confidence interval:
##  62.52341 67.47659
## sample estimates:
## mean of x
##        65
```

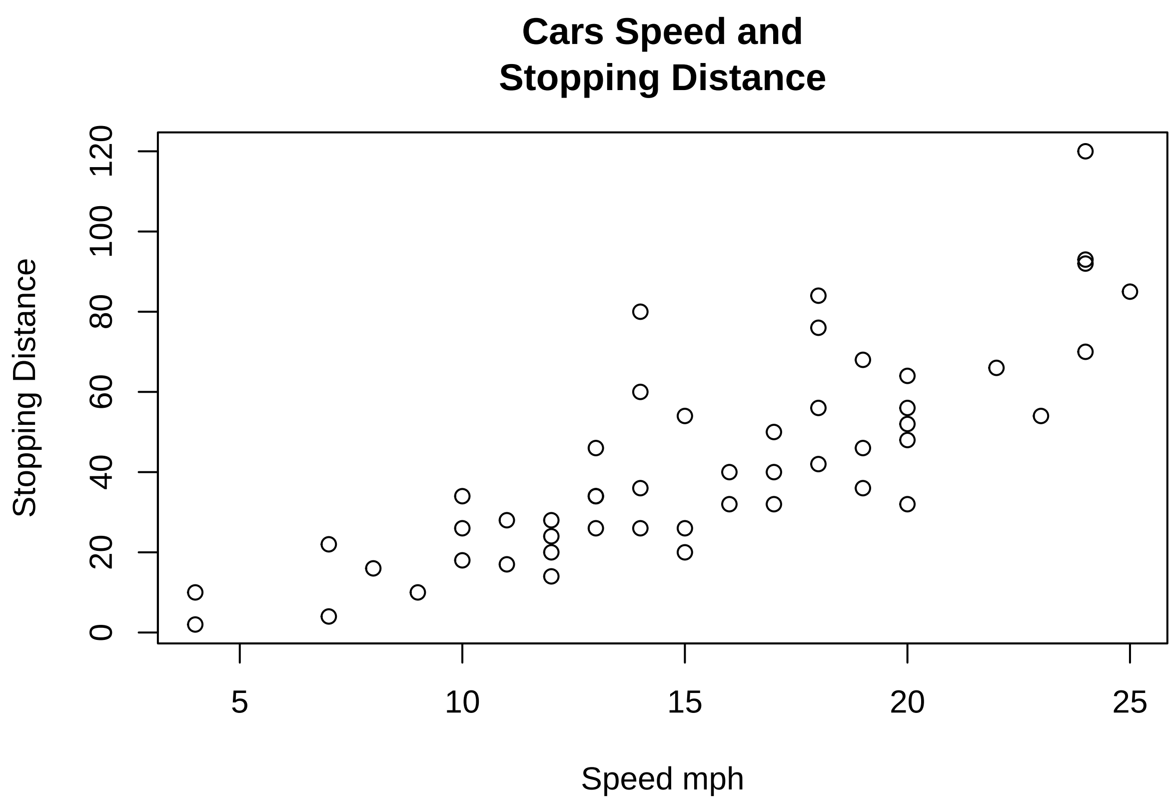
```r
# Interpretation:
# When the null value lies inside the 95% CI, the p-value is large,
# and we fail to reject the null hypothesis.
# When the null value lies outside the CI, the p-value is small,
# and we reject the null hypothesis.

# Relationship:
# A 95% confidence interval contains all the values of mu for which we would not reject H0
# in a two-sided t-test at significance level alpha = 0.05.
```

**Regressions: Using built in "cars" dataset** #Scatterplot Plus Regression

```
cars_df<-cars
head(cars_df)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
#Scatter plot of car speed and distance

plot(cars$speed,cars$dist,xlab="Speed mph",ylab="Stopping Distance",main = "Cars Speed and
Stopping Distance")
```




```
#Correlation for speed and distance
cor(cars$speed,cars$dist)
```

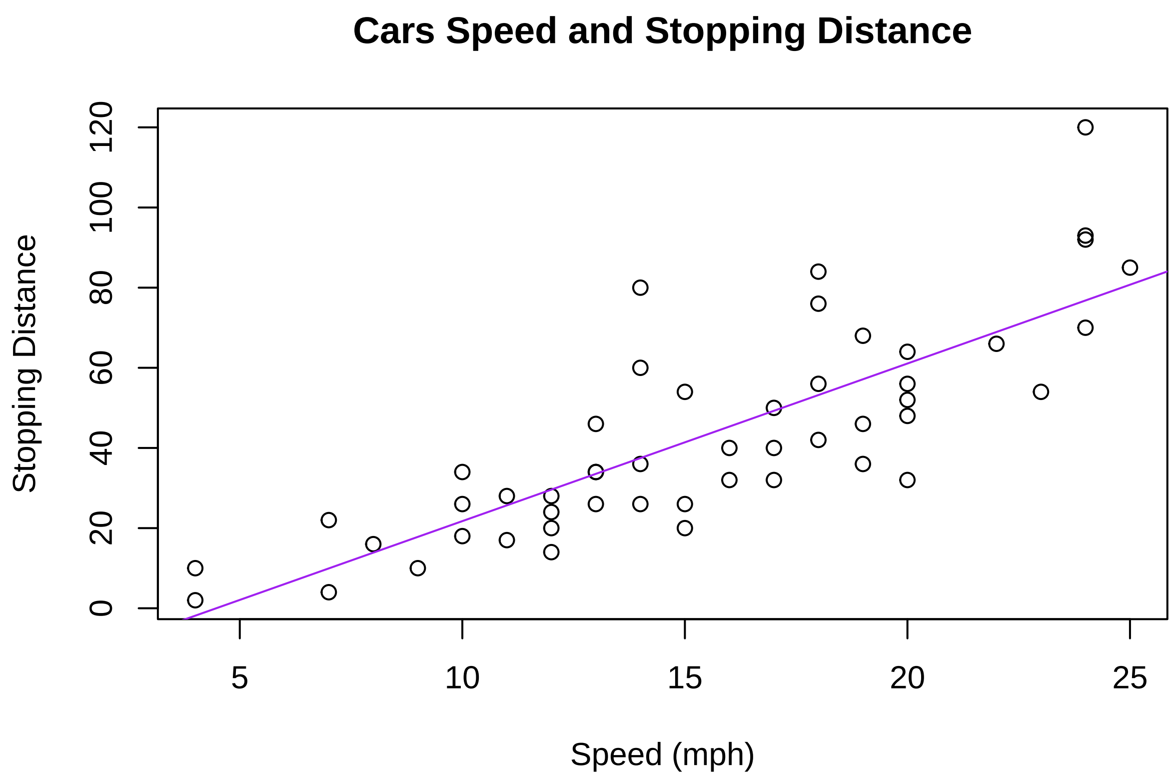
```
## [1] 0.8068949
```

#Regression

```r
# Fit linear regression model
fit <- lm(cars$dist ~ cars$speed)

# View regression summary
summary(fit)
```

```
##
## Call:
## lm(formula = cars$dist ~ cars$speed)
##
## 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 *
## cars$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
```

```r
# Scatter plot with regression line
plot(cars$speed, cars$dist,
     xlab = "Speed (mph)",
     ylab = "Stopping Distance",
     main = "Cars Speed and Stopping Distance")

abline(fit, col = "purple")  # Add regression line
```




```
# To view residuals: did not run because it will be for each data point
#residuals(fit)
```