

Experiment-1- Visualizing Data using R

siju.swamy@saintgits.org

07/07/2021

Tables, charts and plots. Visualising Measures of Central Tendency, Variation, and Shape. Box plots, Pareto diagrams. How to find the mean, median, standard deviation and quantiles of a set of observations. Students may experiment with real as well as artificial data sets.

Tables

Tables can be created from given data. An example is shown bellow: Let's work with the builtin data set iris.

iris						
##	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	
## 7	4.6	3.4	1.4	0.3	setosa	
## 8	5.0	3.4	1.5	0.2	setosa	
## 9	4.4	2.9	1.4	0.2	setosa	
## 10	4.9	3.1	1.5	0.1	setosa	
## 11	5.4	3.7	1.5	0.2	setosa	
## 12	4.8	3.4	1.6	0.2	setosa	
## 13	4.8	3.0	1.4	0.1	setosa	
## 14	4.3	3.0	1.1	0.1	setosa	
## 15	5.8	4.0	1.2	0.2	setosa	
## 16	5.7	4.4	1.5	0.4	setosa	
## 17	5.4	3.9	1.3	0.4	setosa	
## 18	5.1	3.5	1.4	0.3	setosa	
## 19	5.7	3.8	1.7	0.3	setosa	
## 20	5.1	3.8	1.5	0.3	setosa	
## 21	5.4	3.4	1.7	0.2	setosa	
## 22	5.1	3.7	1.5	0.4	setosa	
## 23	4.6	3.6	1.0	0.2	setosa	
## 24	5.1	3.3	1.7	0.5	setosa	
## 25	4.8	3.4	1.9	0.2	setosa	
## 26	5.0	3.0	1.6	0.2	setosa	
## 27	5.0	3.4	1.6	0.4	setosa	
## 28	5.2	3.5	1.5	0.2	setosa	
## 29	5.2	3.4	1.4	0.2	setosa	
## 30	4.7	3.2	1.6	0.2	setosa	
## 31	4.8	3.1	1.6	0.2	setosa	
## 32	5.4	3.4	1.5	0.4	setosa	

## 33	5.2	4.1	1.5	0.1	setosa
## 34	5.5	4.2	1.4	0.2	setosa
## 35	4.9	3.1	1.5	0.2	setosa
## 36	5.0	3.2	1.2	0.2	setosa
## 37	5.5	3.5	1.3	0.2	setosa
## 38	4.9	3.6	1.4	0.1	setosa
## 39	4.4	3.0	1.3	0.2	setosa
## 40	5.1	3.4	1.5	0.2	setosa
## 41	5.0	3.5	1.3	0.3	setosa
## 42	4.5	2.3	1.3	0.3	setosa
## 43	4.4	3.2	1.3	0.2	setosa
## 44	5.0	3.5	1.6	0.6	setosa
## 45	5.1	3.8	1.9	0.4	setosa
## 46	4.8	3.0	1.4	0.3	setosa
## 47	5.1	3.8	1.6	0.2	setosa
## 48	4.6	3.2	1.4	0.2	setosa
## 49	5.3	3.7	1.5	0.2	setosa
## 50	5.0	3.3	1.4	0.2	setosa
## 51	7.0	3.2	4.7	1.4	versicolor
## 52	6.4	3.2	4.5	1.5	versicolor
## 53	6.9	3.1	4.9	1.5	versicolor
## 54	5.5	2.3	4.0	1.3	versicolor
## 55	6.5	2.8	4.6	1.5	versicolor
## 56	5.7	2.8	4.5	1.3	versicolor
## 57	6.3	3.3	4.7	1.6	versicolor
## 58	4.9	2.4	3.3	1.0	versicolor
## 59	6.6	2.9	4.6	1.3	versicolor
## 60	5.2	2.7	3.9	1.4	versicolor
## 61	5.0	2.0	3.5	1.0	versicolor
## 62	5.9	3.0	4.2	1.5	versicolor
## 63	6.0	2.2	4.0	1.0	versicolor
## 64	6.1	2.9	4.7	1.4	versicolor
## 65	5.6	2.9	3.6	1.3	versicolor
## 66	6.7	3.1	4.4	1.4	versicolor
## 67	5.6	3.0	4.5	1.5	versicolor
## 68	5.8	2.7	4.1	1.0	versicolor
## 69	6.2	2.2	4.5	1.5	versicolor
## 70	5.6	2.5	3.9	1.1	versicolor
## 71	5.9	3.2	4.8	1.8	versicolor
## 72	6.1	2.8	4.0	1.3	versicolor
## 73	6.3	2.5	4.9	1.5	versicolor
## 74	6.1	2.8	4.7	1.2	versicolor
## 75	6.4	2.9	4.3	1.3	versicolor
## 76	6.6	3.0	4.4	1.4	versicolor
## 77	6.8	2.8	4.8	1.4	versicolor
## 78	6.7	3.0	5.0	1.7	versicolor
## 79	6.0	2.9	4.5	1.5	versicolor
## 80	5.7	2.6	3.5	1.0	versicolor
## 81	5.5	2.4	3.8	1.1	versicolor
## 82	5.5	2.4	3.7	1.0	versicolor
## 83	5.8	2.7	3.9	1.2	versicolor
## 84	6.0	2.7	5.1	1.6	versicolor
## 85	5.4	3.0	4.5	1.5	versicolor
## 86	6.0	3.4	4.5	1.6	versicolor

## 87	6.7	3.1	4.7	1.5 versicolor
## 88	6.3	2.3	4.4	1.3 versicolor
## 89	5.6	3.0	4.1	1.3 versicolor
## 90	5.5	2.5	4.0	1.3 versicolor
## 91	5.5	2.6	4.4	1.2 versicolor
## 92	6.1	3.0	4.6	1.4 versicolor
## 93	5.8	2.6	4.0	1.2 versicolor
## 94	5.0	2.3	3.3	1.0 versicolor
## 95	5.6	2.7	4.2	1.3 versicolor
## 96	5.7	3.0	4.2	1.2 versicolor
## 97	5.7	2.9	4.2	1.3 versicolor
## 98	6.2	2.9	4.3	1.3 versicolor
## 99	5.1	2.5	3.0	1.1 versicolor
## 100	5.7	2.8	4.1	1.3 versicolor
## 101	6.3	3.3	6.0	2.5 virginica
## 102	5.8	2.7	5.1	1.9 virginica
## 103	7.1	3.0	5.9	2.1 virginica
## 104	6.3	2.9	5.6	1.8 virginica
## 105	6.5	3.0	5.8	2.2 virginica
## 106	7.6	3.0	6.6	2.1 virginica
## 107	4.9	2.5	4.5	1.7 virginica
## 108	7.3	2.9	6.3	1.8 virginica
## 109	6.7	2.5	5.8	1.8 virginica
## 110	7.2	3.6	6.1	2.5 virginica
## 111	6.5	3.2	5.1	2.0 virginica
## 112	6.4	2.7	5.3	1.9 virginica
## 113	6.8	3.0	5.5	2.1 virginica
## 114	5.7	2.5	5.0	2.0 virginica
## 115	5.8	2.8	5.1	2.4 virginica
## 116	6.4	3.2	5.3	2.3 virginica
## 117	6.5	3.0	5.5	1.8 virginica
## 118	7.7	3.8	6.7	2.2 virginica
## 119	7.7	2.6	6.9	2.3 virginica
## 120	6.0	2.2	5.0	1.5 virginica
## 121	6.9	3.2	5.7	2.3 virginica
## 122	5.6	2.8	4.9	2.0 virginica
## 123	7.7	2.8	6.7	2.0 virginica
## 124	6.3	2.7	4.9	1.8 virginica
## 125	6.7	3.3	5.7	2.1 virginica
## 126	7.2	3.2	6.0	1.8 virginica
## 127	6.2	2.8	4.8	1.8 virginica
## 128	6.1	3.0	4.9	1.8 virginica
## 129	6.4	2.8	5.6	2.1 virginica
## 130	7.2	3.0	5.8	1.6 virginica
## 131	7.4	2.8	6.1	1.9 virginica
## 132	7.9	3.8	6.4	2.0 virginica
## 133	6.4	2.8	5.6	2.2 virginica
## 134	6.3	2.8	5.1	1.5 virginica
## 135	6.1	2.6	5.6	1.4 virginica
## 136	7.7	3.0	6.1	2.3 virginica
## 137	6.3	3.4	5.6	2.4 virginica
## 138	6.4	3.1	5.5	1.8 virginica
## 139	6.0	3.0	4.8	1.8 virginica
## 140	6.9	3.1	5.4	2.1 virginica

```
## 141      6.7      3.1      5.6      2.4 virginica
## 142      6.9      3.1      5.1      2.3 virginica
## 143      5.8      2.7      5.1      1.9 virginica
## 144      6.8      3.2      5.9      2.3 virginica
## 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
```

```
## Frequency table with table() function in R
table(iris$Sepal.Length)
```

```
##
## 4.3 4.4 4.5 4.6 4.7 4.8 4.9 5 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9 6
## 1 3 1 4 2 5 6 10 9 4 1 6 7 6 8 7 3 6
## 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7 7.1 7.2 7.3 7.4 7.6 7.7 7.9
## 6 4 9 7 5 2 8 3 4 1 1 3 1 1 1 4 1
```

```
table(iris$Species,iris$Sepal.Length)
```

```
##
##           4.3 4.4 4.5 4.6 4.7 4.8 4.9 5 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8
## setosa      1 3 1 4 2 5 4 8 8 3 1 5 2 0 2 1
## versicolor  0 0 0 0 0 0 1 2 1 1 0 1 5 5 5 3
## virginica   0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 3
##
##           5.9 6 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7 7.1 7.2 7.3 7.4
## setosa      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## versicolor  2 4 4 2 3 2 1 2 3 1 1 1 0 0 0 0
## virginica   1 2 2 2 6 5 4 0 5 2 3 0 1 3 1 1
##
##           7.6 7.7 7.9
## setosa      0 0 0
## versicolor  0 0 0
## virginica   1 4 1
```

Frequency Table with Proportion:

proportion of the frequency table is created using prop.table() function. Table is passed as an argument to the prop.table() function. so that the proportion of the frequency table is calculated

```
## Frequency table with with proportion using table() function in R
table1 = table(iris$Species)
prop.table(table1)*100
```

```
##
## setosa versicolor virginica
## 33.33333 33.33333 33.33333
```

Frequency table with condition:

We can also create a frequency table with predefined condition using R table() function. For example let's say we need to get how many observations have Sepal.Length > 1.0 in iris table.

```
table(iris$Sepal.Length>=5.0)
```

```
##
## FALSE TRUE
##    22   128
```

2 way cross table in R:

Table function also helpful in creating 2 way cross table in R. For example lets say we need to create cross tabulation of gears and carb in mtcars table

```
mtcars
```

##	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
## Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
## Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
## Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
## Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
## Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
## Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
## Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
## Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
## Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
## Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
## Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
## Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
## Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
## Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
## Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
## Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
## AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
## Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
## Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
## Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
## Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
## Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
## Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
## Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
## Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
## Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

```
table(mtcars$gear,mtcars$carb)
```

```
##
##    1 2 3 4 6 8
##    3 3 4 3 5 0 0
##    4 4 4 0 4 0 0
##    5 0 2 0 1 1 1
```

3 way cross table in R:

Similar to 2 way cross table we can create a 3 way cross table in R with the help of table function.

```
table(mtcars$gear,mtcars$carb,mtcars$cyl)
```

```
## , , = 4
##
##
##      1 2 3 4 6 8
##    3 1 0 0 0 0
##    4 4 4 0 0 0
##    5 0 2 0 0 0
##
## , , = 6
##
##
##      1 2 3 4 6 8
##    3 2 0 0 0 0
##    4 0 0 0 4 0
##    5 0 0 0 0 1
##
## , , = 8
##
##
##      1 2 3 4 6 8
##    3 0 4 3 5 0
##    4 0 0 0 0 0
##    5 0 0 0 1 0
```

Data visialization in r

The most popular visualization library in R programming is ggplot2- *grammar of graphics for plots*

```
#set.seed(123)
sample(nrow(iris),10)
```

```
## [1] 76 18 54 61 84 9 46 106 11 38
```

```
library(ggplot2) # Data visualization
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 3.5.3
```

```
library(grid)
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.5.3
```

```
iris=iris
# First let's get a random sampling of the data
iris[sample(nrow(iris),10),]
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 7              4.6          3.4          1.4          0.3    setosa
## 134             6.3          2.8          5.1          1.5  virginica
## 89              5.6          3.0          4.1          1.3  versicolor
## 98              6.2          2.9          4.3          1.3  versicolor
```

## 141	6.7	3.1	5.6	2.4	virginica
## 73	6.3	2.5	4.9	1.5	versicolor
## 138	6.4	3.1	5.5	1.8	virginica
## 111	6.5	3.2	5.1	2.0	virginica
## 5	5.0	3.6	1.4	0.2	setosa
## 122	5.6	2.8	4.9	2.0	virginica

Density plots

```
# Density & Frequency analysis with the Histogram,

# Sepal length
HisSl <- ggplot(data=iris, aes(x=Sepal.Length))+
  geom_histogram(binwidth=0.2, color="black", aes(fill=Species)) +
  xlab("Sepal Length (cm)") +
  ylab("Frequency") +
  theme(legend.position="none")+
  ggtitle("Histogram of Sepal Length")+
  geom_vline(data=iris, aes(xintercept = mean(Sepal.Length)), linetype="dashed", color="black")
#HisSl

# Sepal width
HistSw <- ggplot(data=iris, aes(x=Sepal.Width)) +
  geom_histogram(binwidth=0.2, color="black", aes(fill=Species)) +
  xlab("Sepal Width (cm)") +
  ylab("Frequency") +
  theme(legend.position="none")+
  ggtitle("Histogram of Sepal Width")+
  geom_vline(data=iris, aes(xintercept = mean(Sepal.Width)), linetype="dashed", color="grey")
# Petal length
HistPl <- ggplot(data=iris, aes(x=Petal.Length))+
  geom_histogram(binwidth=0.2, color="black", aes(fill=Species)) +
  xlab("Petal Length (cm)") +
  ylab("Frequency") +
  theme(legend.position="none")+
  ggtitle("Histogram of Petal Length")+
  geom_vline(data=iris, aes(xintercept = mean(Petal.Length)),
    linetype="dashed", color="grey")

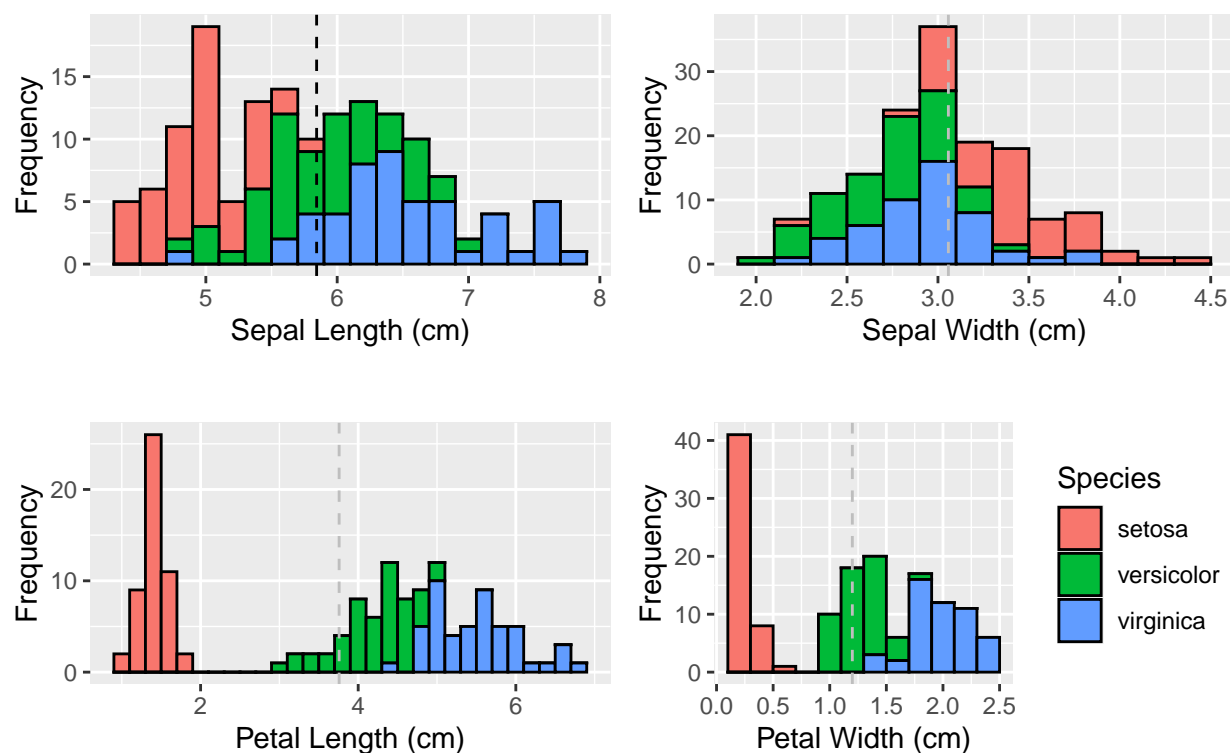
# Petal width
HistPw <- ggplot(data=iris, aes(x=Petal.Width))+
  geom_histogram(binwidth=0.2, color="black", aes(fill=Species)) +
  xlab("Petal Width (cm)") +
  ylab("Frequency") +
  theme(legend.position="right") +
  ggtitle("Histogram of Petal Width")+
  geom_vline(data=iris, aes(xintercept = mean(Petal.Width)), linetype="dashed", color="grey")
```

Creating the plots

```
# Plot all visualizations
grid.arrange(HisSl + ggtitle(""),
```

```
HistSw + ggtitle(""),
HistPl + ggtitle(""),
HistPw + ggtitle(""),
nrow = 2,
top = textGrob("Iris Frequency Histogram",
               gp=gpar(fontsize=15))
)
```

Iris Frequency Histogram



Creating Distribution plots in R

*# Notice the shape of the data, most attributes exhibit a normal distribution.
You can see the measurements of very small flowers in the Petal width and length column.*

*# We can review the density distribution of each attribute broken down by class value.
Like the scatterplot matrix, the density plot by class can help see the separation of classes.
It can also help to understand the overlap in class values for an attribute.*

```
DhistPl <- ggplot(iris, aes(x=Petal.Length, colour=Species, fill=Species)) +
  geom_density(alpha=.3) +
  geom_vline(aes(xintercept=mean(Petal.Length), colour=Species), linetype="dashed", color="grey", size=1) +
  xlab("Petal Length (cm)") +
  ylab("Density") +
  theme(legend.position="none")
```



```

DhistPw <- ggplot(iris, aes(x=Petal.Width, colour=Species, fill=Species)) +
  geom_density(alpha=.3) +
  geom_vline(aes(xintercept=mean(Petal.Width), colour=Species), linetype="dashed", color="grey", size=1) +
  xlab("Petal Width (cm)") +
  ylab("Density")

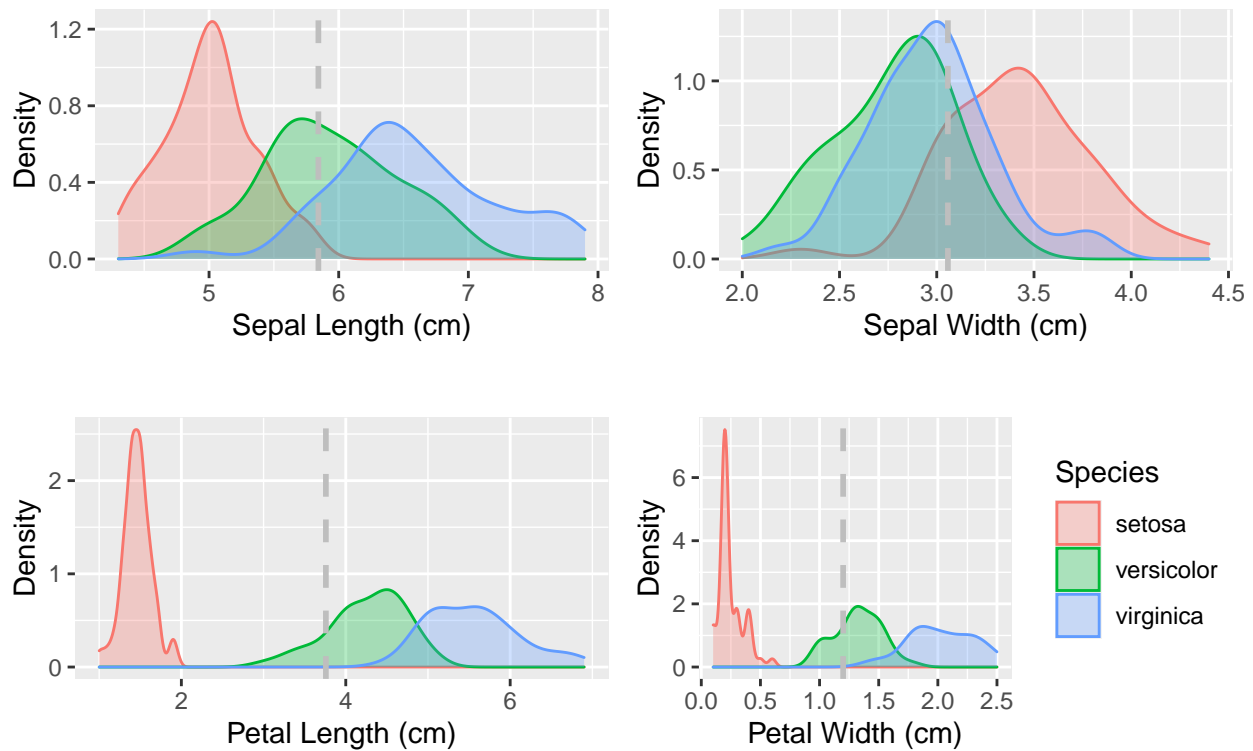
DhistSw <- ggplot(iris, aes(x=Sepal.Width, colour=Species, fill=Species)) +
  geom_density(alpha=.3) +
  geom_vline(aes(xintercept=mean(Sepal.Width), colour=Species), linetype="dashed", color="grey", size=1) +
  xlab("Sepal Width (cm)") +
  ylab("Density") +
  theme(legend.position="none")

DhistSl <- ggplot(iris, aes(x=Sepal.Length, colour=Species, fill=Species)) +
  geom_density(alpha=.3) +
  geom_vline(aes(xintercept=mean(Sepal.Length), colour=Species), linetype="dashed", color="grey", size=1) +
  xlab("Sepal Length (cm)") +
  ylab("Density") +
  theme(legend.position="none")

# Plot all density visualizations
grid.arrange(DhistSl + ggtitle(""),
              DhistSw + ggtitle(""),
              DhistPl + ggtitle(""),
              DhistPw + ggtitle(""),
              nrow = 2,
              top = textGrob("Iris Density Plot",
                             gp=gpar(fontsize=15))
)

```

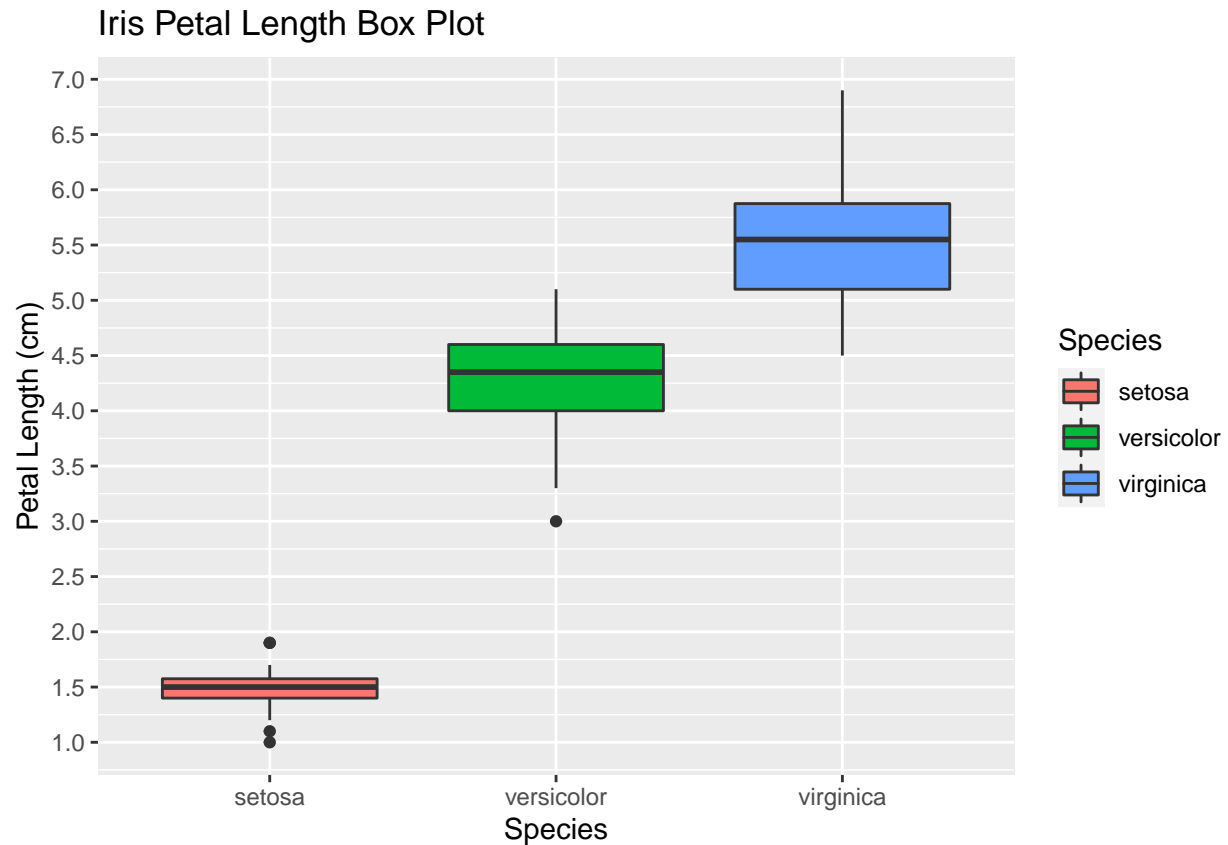
Iris Density Plot



Box plot to visualize variation

Next with the boxplot we will identify some outliers. As you can see some classes do not overlap at all (Petal Length) where as with other attributes there are hard to tease apart (Sepal Width).

```
ggplot(iris, aes(Species, Petal.Length, fill=Species)) +
  geom_boxplot() +
  scale_y_continuous("Petal Length (cm)", breaks= seq(0,30, by=.5)) +
  labs(title = "Iris Petal Length Box Plot", x = "Species")
```



Grid-plot of Box plots

Let's plot all the variables in a single visualization that will contain all the boxplots

```
BpSl <- ggplot(iris, aes(Species, Sepal.Length, fill=Species)) +
  geom_boxplot()+
  scale_y_continuous("Sepal Length (cm)", breaks= seq(0,30, by=.5))+
  theme(legend.position="none")

BpSw <- ggplot(iris, aes(Species, Sepal.Width, fill=Species)) +
  geom_boxplot()+
  scale_y_continuous("Sepal Width (cm)", breaks= seq(0,30, by=.5))+
  theme(legend.position="none")

BpPl <- ggplot(iris, aes(Species, Petal.Length, fill=Species)) +
  geom_boxplot()+
  scale_y_continuous("Petal Length (cm)", breaks= seq(0,30, by=.5))+
  theme(legend.position="none")
```

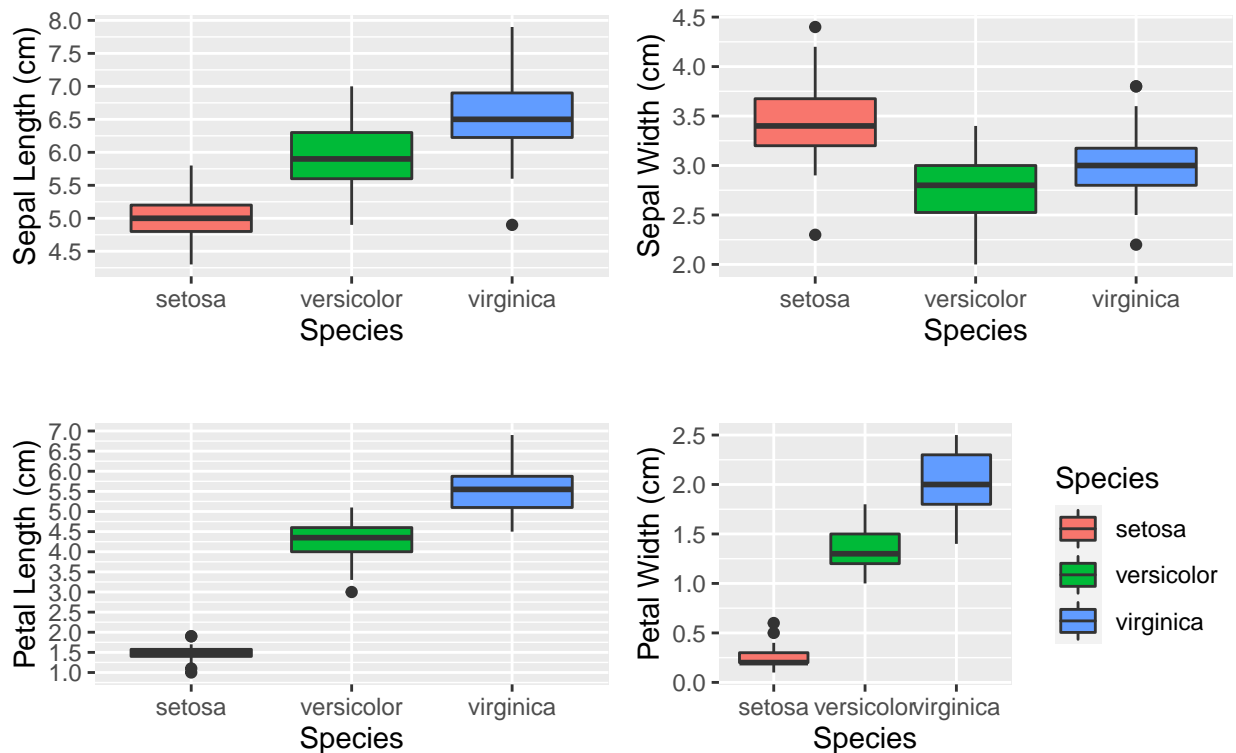
```

BpPw <- ggplot(iris, aes(Species, Petal.Width, fill=Species)) +
  geom_boxplot()+
  scale_y_continuous("Petal Width (cm)", breaks= seq(0,30, by=.5))+
  labs(title = "Iris Box Plot", x = "Species")

# Plot all visualizations
grid.arrange(BpSl + ggtitle(""),
  BpSw + ggtitle(""),
  BpPl + ggtitle(""),
  BpPw + ggtitle(""),
  nrow = 2,
  top = textGrob("Sepal and Petal Box Plot",
    gp=gpar(fontsize=15))
)

```

Sepal and Petal Box Plot



Violin plot- Another visualization method

Violin plots are an alternative to box plots that solves the issues regarding displaying the underlying distribution of the observations, as these plots show a kernel density estimate of the data. In this tutorial, we will show you how to create a violin plot in base R from a vector and from data frames, how to add mean points and split the R violin plots by group. They show the number of points at a particular value by the width of the shapes. They can also include the marker for the median and a box for the interquartile range.

```

VpSl <- ggplot(iris, aes(Species, Sepal.Length, fill=Species)) +
  geom_violin(aes(color = Species), trim = T)+
  scale_y_continuous("Sepal Length", breaks= seq(0,30, by=.5))+
  geom_boxplot(width=0.1)+
  theme(legend.position="none")

VpSw <- ggplot(iris, aes(Species, Sepal.Width, fill=Species)) +
  geom_violin(aes(color = Species), trim = T)+
  scale_y_continuous("Sepal Width", breaks= seq(0,30, by=.5))+
  geom_boxplot(width=0.1)+
  theme(legend.position="none")

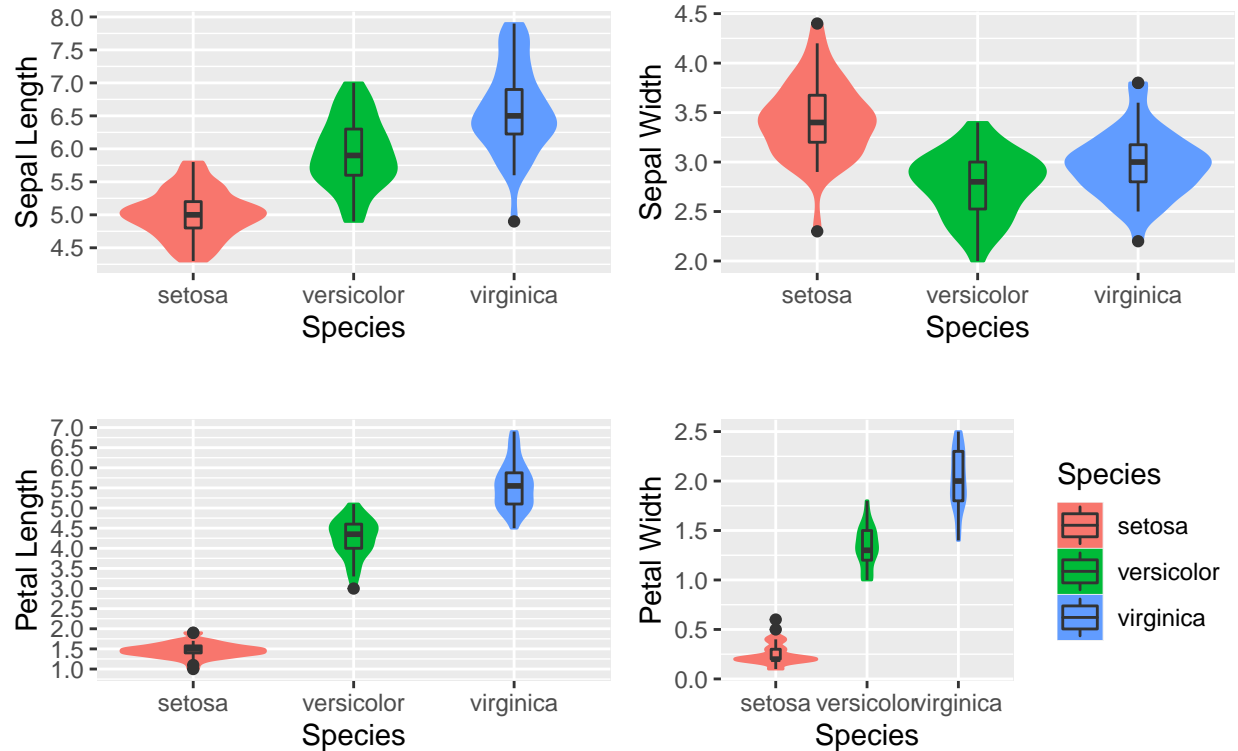
VpPl <- ggplot(iris, aes(Species, Petal.Length, fill=Species)) +
  geom_violin(aes(color = Species), trim = T)+
  scale_y_continuous("Petal Length", breaks= seq(0,30, by=.5))+
  geom_boxplot(width=0.1)+
  theme(legend.position="none")

VpPw <- ggplot(iris, aes(Species, Petal.Width, fill=Species)) +
  geom_violin(aes(color = Species), trim = T)+
  scale_y_continuous("Petal Width", breaks= seq(0,30, by=.5))+
  geom_boxplot(width=0.1)+
  labs(title = "Iris Box Plot", x = "Species")

# Plot all visualizations
grid.arrange(VpSl + ggtitle(""),
             VpSw + ggtitle(""),
             VpPl + ggtitle(""),
             VpPw + ggtitle(""),
             nrow = 2,
             top = textGrob("Sepal and Petal Violin Plot",
                           gp=gpar(fontsize=15))
)

```

Sepal and Petal Violin Plot



Scatter plots

Now let's create a scatterplot of petal lengths versus petal widths with the color & shape by species. There is also a regression line with a 95% confidence band. Notice the petal length of the setosa is clearly a differentiated cluster so it will be a good predictor for ML.

```
ggplot(data = iris, aes(x = Petal.Length, y = Petal.Width)) +
  xlab("Petal Length") +
  ylab("Petal Width") +
  geom_point(aes(color = Species, shape = Species)) +
  geom_smooth(method = 'lm') +
  ggtitle("Petal Length vs Width")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

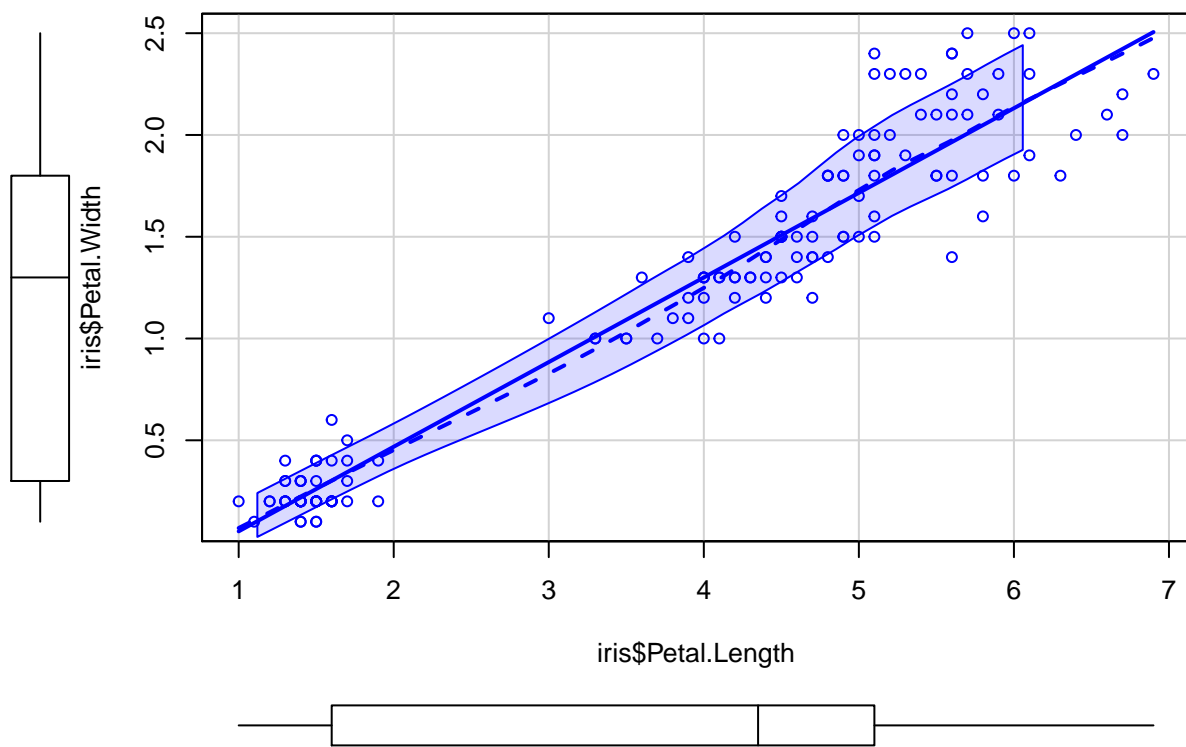


Here is a similar plot with more details on the regression line.

```
library(car)
```

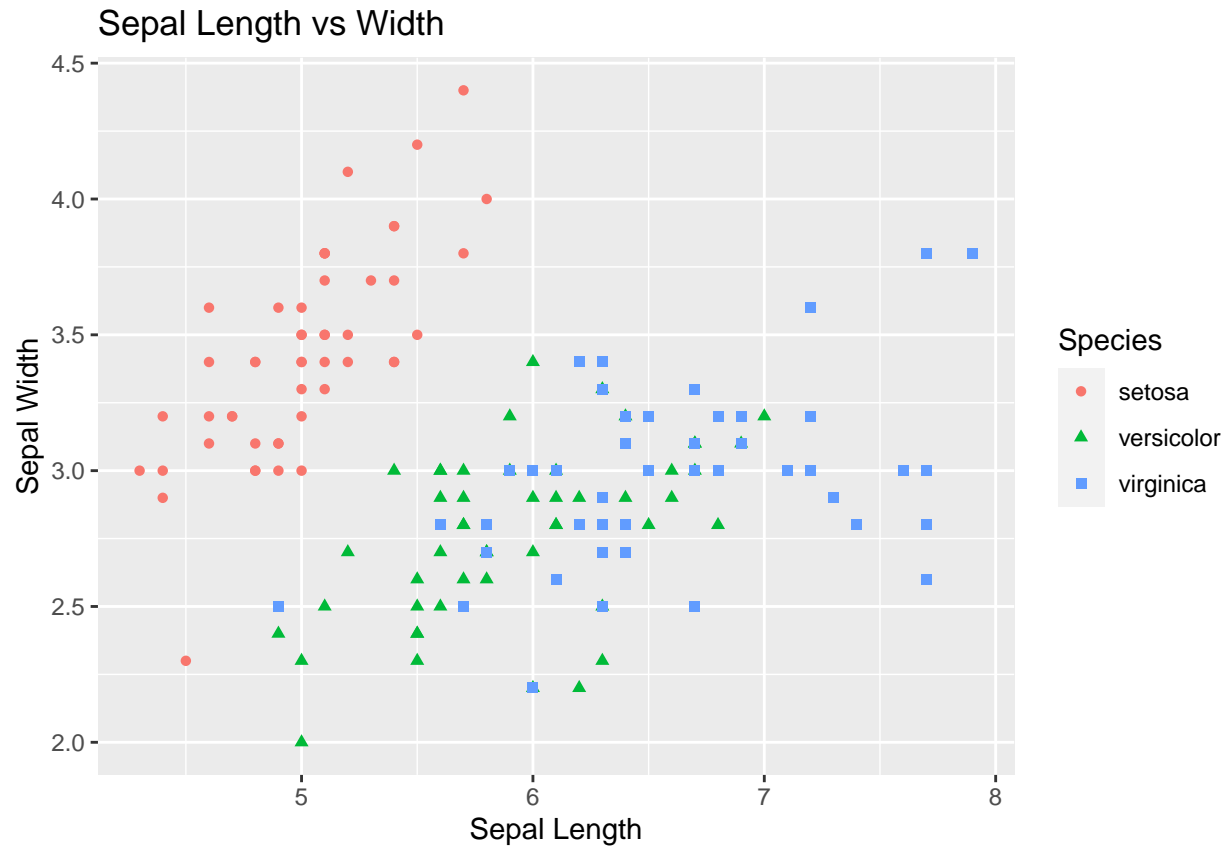
```
## Loading required package: carData
```

```
scatterplot(iris$Petal.Length,iris$Petal.Width)
```



Now check the Sepal Length vs Width. Notice the sepal of the Virginica and Versicolor species is more

```
ggplot(data=iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(aes(color=Species, shape=Species)) +
  xlab("Sepal Length") +
  ylab("Sepal Width") +
  ggtitle("Sepal Length vs Width")
```

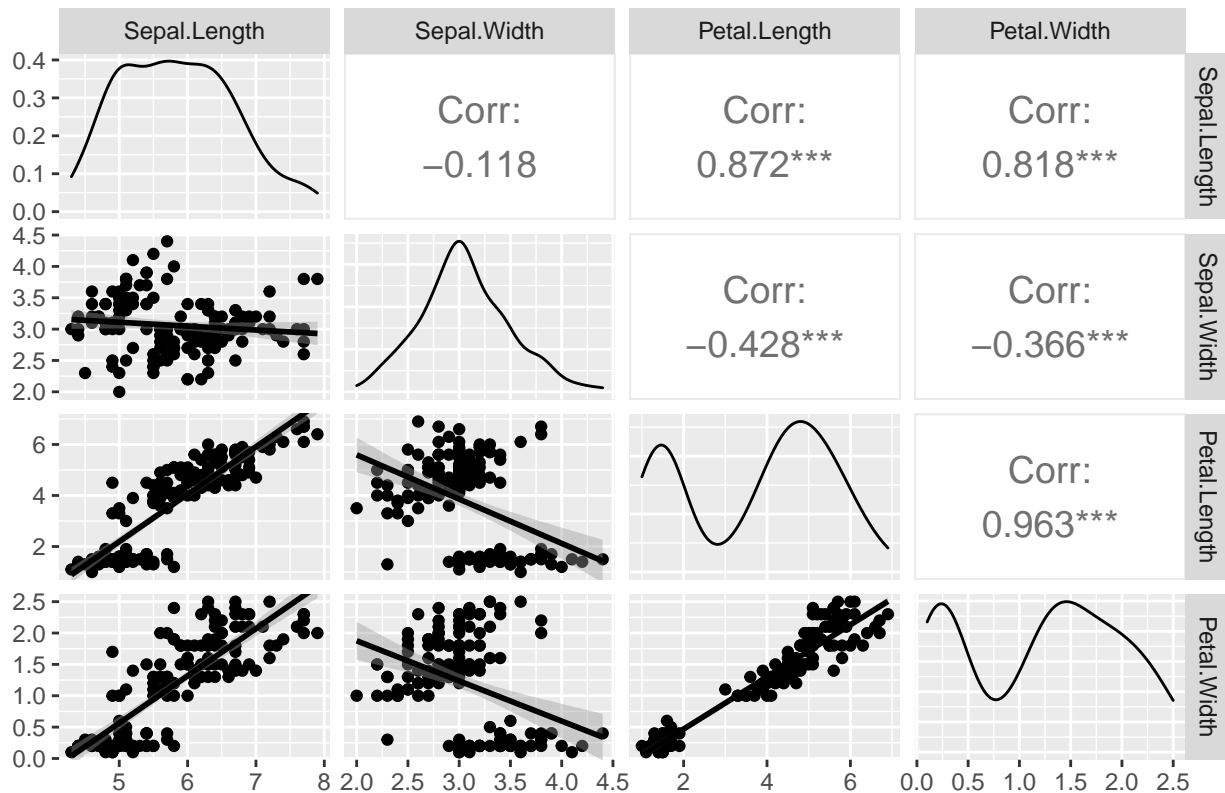



Correlation Plots

Based on all the plots we have done we can see there is certain correlation. Let's take a look at the pairwise correlation numerical values to ascertain the relationships in more detail.

```
library(GGally)
ggpairs(data = iris[1:4],
  title = "Iris Correlation Plot",
  upper = list(continuous = wrap("cor", size = 5)),
  lower = list(continuous = "smooth")
)
```

Iris Correlation Plot



Pareto Charts

A Pareto chart is a bar graph. The lengths of the bars represent frequency or cost (time or money), and are arranged with longest bars on the left and the shortest to the right. In this way the chart visually depicts which situations are more significant. This cause analysis tool is considered one of the seven basic quality tools.

WHEN TO USE A PARETO CHART

1. When analyzing data about the frequency of problems or causes in a process
2. When there are many problems or causes and you want to focus on the most significant
3. When analyzing broad causes by looking at their specific components
4. When communicating with others about your data

```
library(ggplot2)
library(ggQC)
```

```
## Warning: package 'ggQC' was built under R version 3.5.3
```

```
Data4Pareto <- data.frame(
  KPI = c("Customer Service Time", "Order Fulfillment", "Order Processing Time",
          "Order Production Time", "Order Quality Control Time", "Rework Time",
          "Shipping"),
  Time = c(1.50, 38.50, 3.75, 23.08, 1.92, 3.58, 73.17))
```

```
ggplot2::ggplot(Data4Pareto, aes(x = KPI, y = Time)) +
  ggQC::stat_pareto(point.color = "red",
    point.size = 3,
    line.color = "black",
    bars.fill = c("blue", "orange")) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust=0.5))
```

