Introduction to Statistics with R

Session R01: Basics and Diagrams

Marvin Schmitt

Prequel: The pipe operator %>%

If you load the `tidyverse` library, you can use the %>% pipe.

Instead of `function_b(function_a(x))` we can write

`x %>% function_a() %>% function_b()`

```r
x = 10
round(log(x), 3)
```

```
## [1] 2.303
```

```r
x %>% log() %>% round(3)
```

```
## [1] 2.303
```

The %>% operator works like a **pipe** and passes the left-hand-side as the first argument to the function on the right-hand-side.

Sampling from a distribution

`rnorm(n=n, mean=mu, sd=sigma)` draws `n` samples from the normal distribution $\mathcal{N}(\mu, \sigma)$:

```r
IQ_values = rnorm(n=200, mean=100, sd=15) %>% round()
print(IQ_values)
```

```
##   [1]  94  94  74 105  79  97 111 108 121  95  81  78 107 129  81  77 105 112
##  [19] 109  87 102  96  92 114 103 113 108  92  93 113 119  80 104  80 124  88
##  [37] 128 102 114  84  92  73 110 103 118 101 106  89  94 109  97  87  88 110
##  [55] 113  71  93  84  95 100 123  92  86 103 120  81 113 134  87 100 116  96
##  [73]  86  88  85  63 106  83 122  93  70 108  97  89 103 117  99 111  67 101
##  [91]  86 121 112  80 107 106  77  94  97 105 110  98  84  92  83  95 124  86
## [109]  95 112  91  92  93  95  97 110 144 108  91  68  95 120  86 112  99  80
## [127]  94 100  99 122  90  77  95 109 100 120 103 124 121  97 130  93 123 101
## [145] 106  85 106 124  75  86 104 127 113  93  98  84 120 137 106 109  68 112
## [163] 112 125 109 112 103  97  87 110 104  89 111 129 113  71  88  99  91  95
## [181]  86  70 108 110 106 111 100 114 133  96 110 112 112  98  82  95 106 125
## [199]  84 105
```

Descriptive statistics

```
mean(IQ_values)          # Mean
```

```
## [1] 100.345
```

```
var(IQ_values)           # Variance
```

```
## [1] 236.9105
```

```
sd(IQ_values)            # Standard Deviation sigma
```

```
## [1] 15.3919
```

```
min(IQ_values)           # Minimum
```

```
## [1] 63
```

```
max(IQ_values)           # Maximum
```

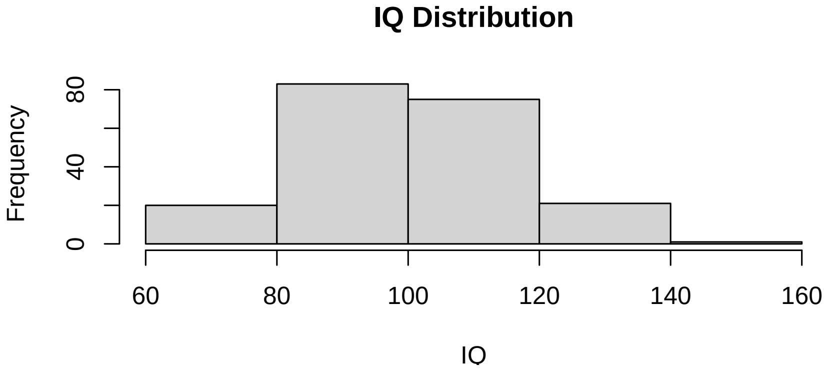
```
## [1] 144
```

```
max(IQ_values) - min(IQ_values)  # Range
```

```
## [1] 81
```

Plots: Histogram in base R

```
hist(IQ_values, main="IQ Distribution", xlab = "IQ", breaks = 5)
```



Plots: ggplot2 introduction

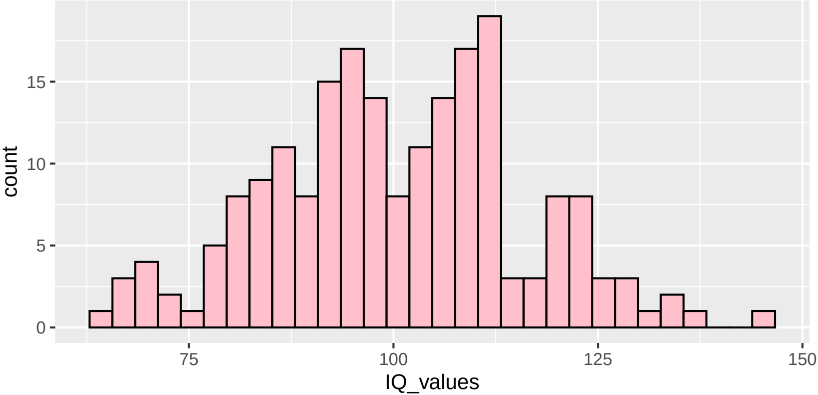
- ggplot2 is a modern library to generate publication-ready plots.
- When you are visualizing data, it should usually be your first choice.
- The ggplot2 syntax is modular and different from the base R syntax. -ggplot2 works best on data frames, so let's turn x into a data frame:

```
df_IQ = data.frame(IQ_values)
```

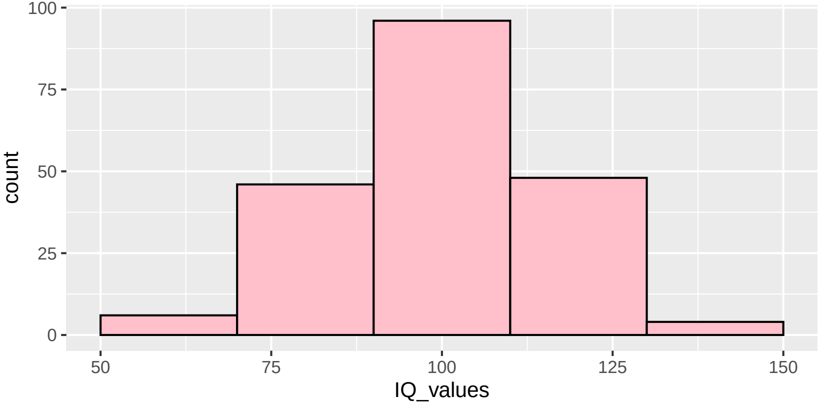
Plots: ggplot2 histogram

```
ggplot(data=df_IQ, aes(x=IQ_values)) +
  geom_histogram(fill="pink", color="black")
```

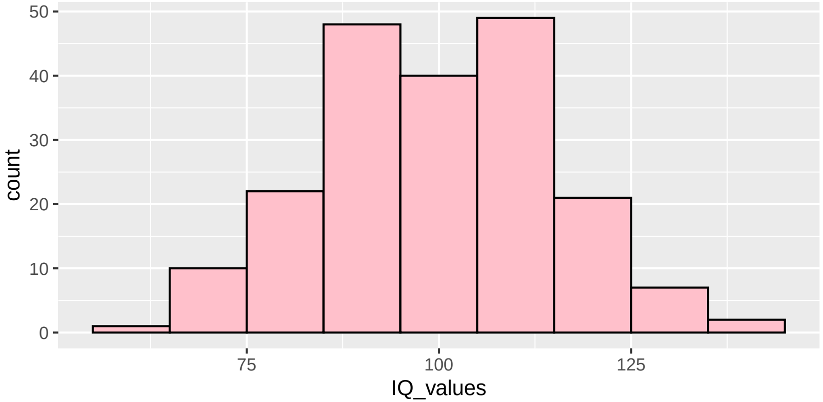
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data=df_IQ, aes(x=IQ_values)) +
  geom_histogram(binwidth=20, fill="pink", color="black")
```

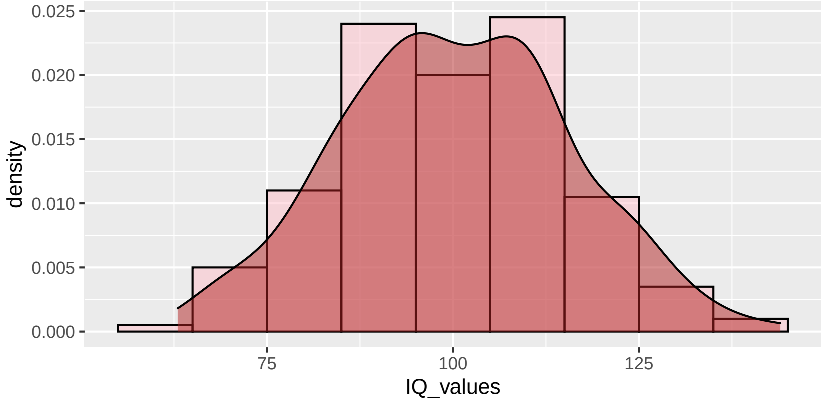


```r
ggplot(data=df_IQ, aes(x=IQ_values)) +
  geom_histogram(binwidth=10, fill="pink", color="black")
```



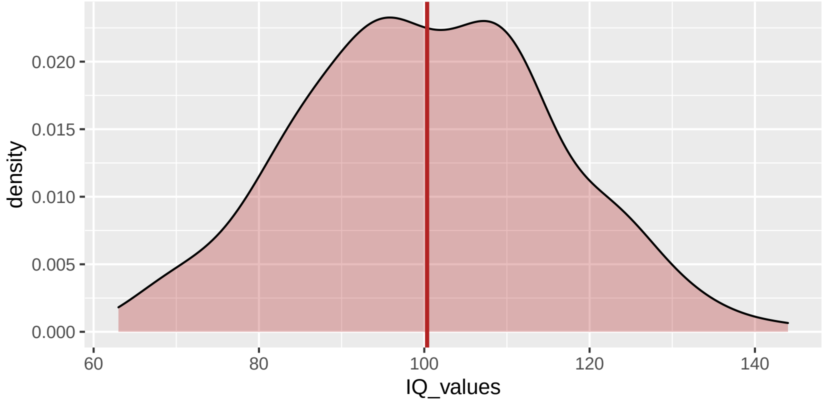
Plots: ggplot2 density plot

```
ggplot(data=df_IQ, aes(x=IQ_values)) +
  geom_histogram(aes(y=..density..), binwidth=10, fill="pink",
                 color="black", alpha=.50) +
  geom_density(fill="firebrick", alpha = .50)
```



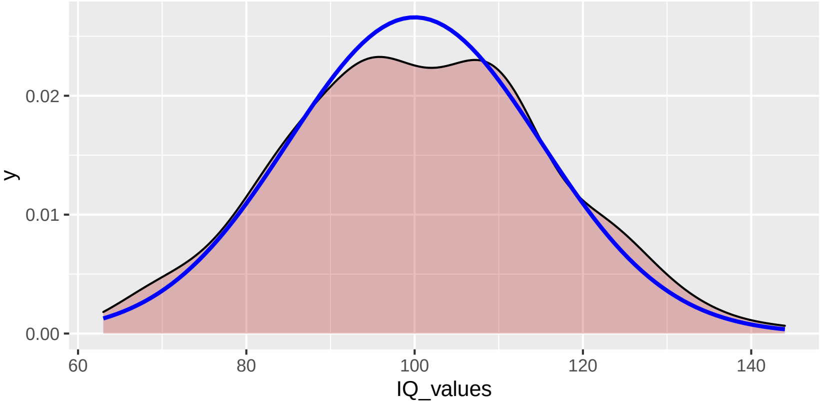
Plots: ggplot2 density plot with mean

```
ggplot(data=df_IQ, aes(x=IQ_values)) +
  geom_density(fill="firebrick", alpha = .30) +
  geom_vline(aes(xintercept = mean(IQ_values)), size=1, col="firebrick")
```



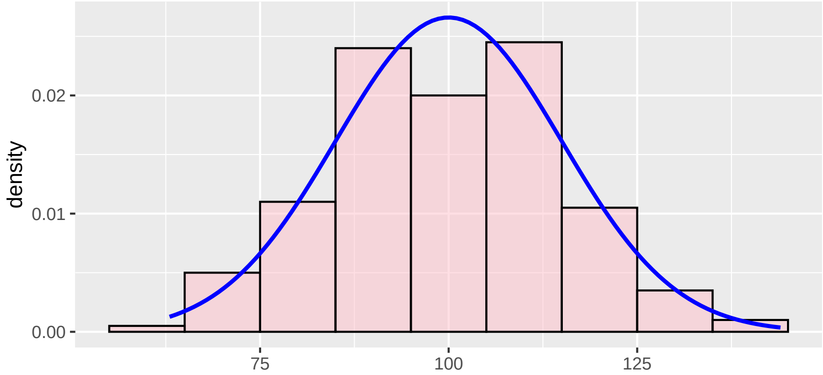
Plots: ggplot2 density plot with normal distribution

```r
ggplot(data=df_IQ, aes(x=IQ_values)) +
  geom_density(fill="firebrick", alpha = .30) +
  stat_function(fun = dnorm, n = 101, args = list(mean = 100, sd = 15),
                col="blue", size=1)
```



Plots: ggplot2 histogram with normal distribution

```
ggplot(data=df_IQ, aes(x=IQ_values)) +
  geom_histogram(aes(y=..density..), binwidth=10, fill="pink",
                 color="black", alpha=.50) +
  stat_function(fun = dnorm, n = 101, args = list(mean = 100, sd = 15),
                col="blue", size=1)
```



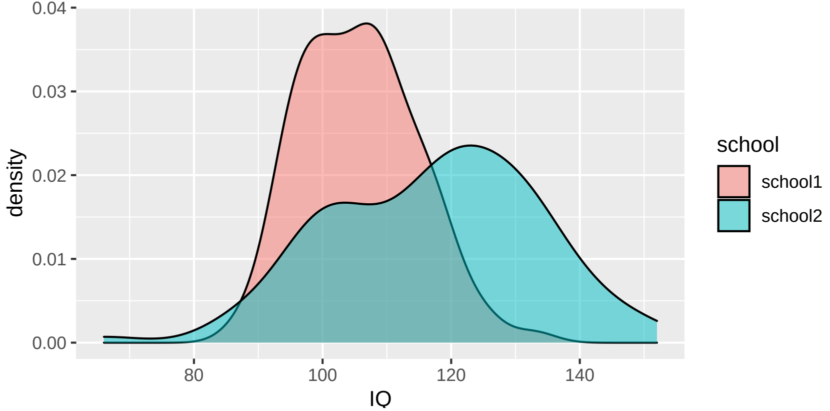
Plots: Plotting different groups

We will add a group variable `school` and simulate data:

```r
IQ_1 = rnorm(n=100, mean=105, sd=10) %>% round()
IQ_2 = rnorm(n=100, mean=120, sd=16) %>% round()
df_1 = data.frame(school="school1", IQ=IQ_1)
df_2 = data.frame(school="school2", IQ=IQ_2)
df_schools = rbind(df_1, df_2)
```

Plots: ggplot2 density plot with 2 groups

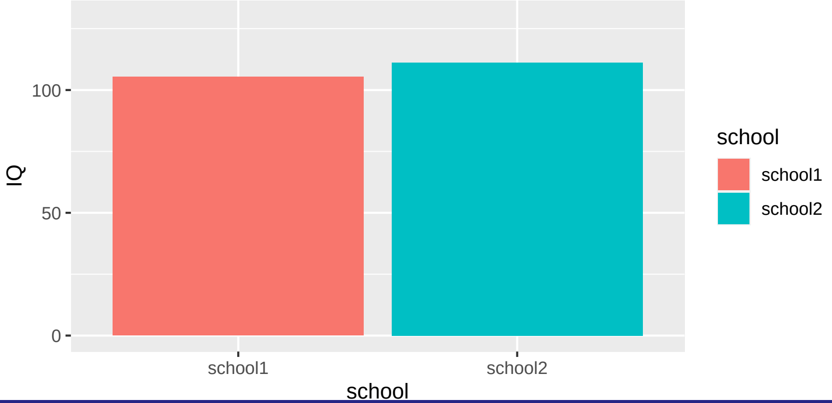
```r
ggplot(data=df_schools, aes(x=IQ, fill=school)) +
  geom_density(alpha = .50)
```



Plots: ggplot2 barplot

```
ggplot(data=df_schools, aes(x=school, y=IQ, fill=school)) +
  geom_bar(stat="summary", fun="mean") +
  ylim(0, 130)
```

```
## Warning: Removed 24 rows containing non-finite values (stat_summary).
```


IQ
school1
school2
school
school
🟥 school1
🟦 school2