Algebra + Data Science R CHEAT SHEET

CourseKata

Word Equations (not R)

outcome = predictor + other stuff

Y = X + other stuff

Basic Scatterplot

```
gf_point(Y ~ X, data = data_set)
```

Data Frame

```
# view first/last six rows
head(data_set)
tail(data_set)
```

Other Basics

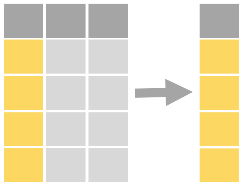
```
print("Hello world!")

# arithmetic operations
sum(1, 2, 100)
+, -, *, /

# logical operations
>, <, >=, <=, ==, !=, |, &
```

Custom Function

```
our_function <- function(X){-5.5 + 49*X}

# evaluate a function
our_function(.24)
```

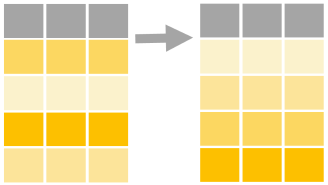
Manipulate Data Frame

```
# select multiple variables
select(data_set, Y1, Y2)
```

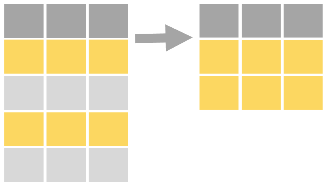


```
# save new data frame
new_data <- select(data_set, Y1, Y2)
```

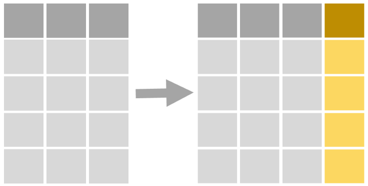
```
# arrange rows by variable
arrange(data_set, Y)
```



```
# find rows that meet condition
filter(data_set, Y > 300)
```



```
# create a new variable in data frame
mutate(data_set, Y3 = Y1 - Y2)
```



Extract and Save a Vector

```
Y <- data_set$variable
```

Predictions and Residuals

```
prediction <- our_function(X)

residual <- Y - our_function(X)

# sum of residuals
sum(residual)

# sum of squared residuals
sum(residual^2)
```

Statistics with Vectors

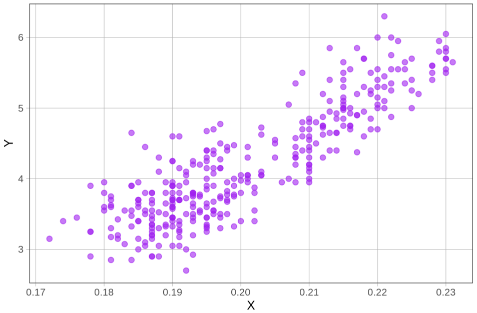
```
mean(Y)

sse(Y, our_function(X))

mse(Y, our_function(X))

rmse(Y, our_function(X))
```

Best-Fitting Linear Model

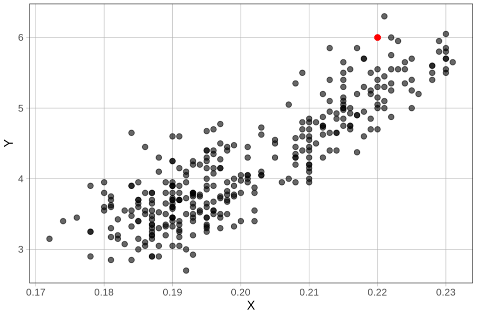
```
# use one explanatory variable
lm(Y ~ X, data = data_set)
```

Visualizations

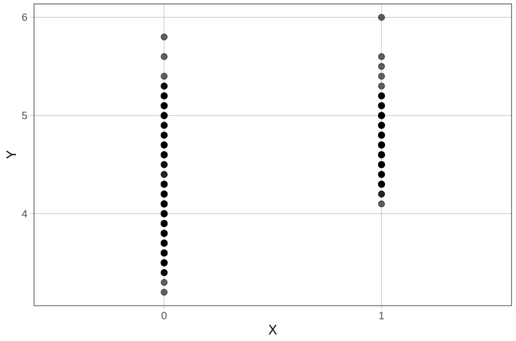
```
# basic scatterplot
gf_point(Y ~ X, data = data_set,
  color = "purple")
```



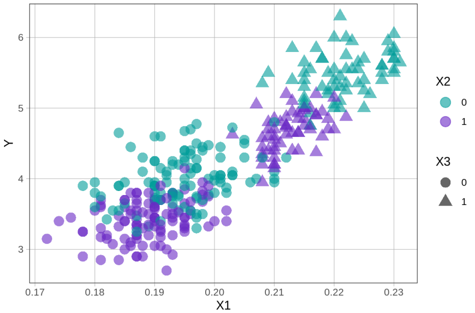
```
# add points to a scatterplot
gf_point(Y ~ X, data = data_set) %>%
  gf_point(6 ~ 0.22, color = "red")
```



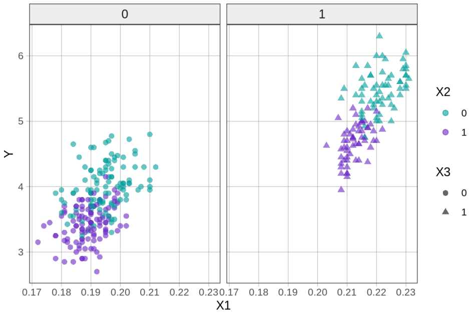
```
# scatterplot with categorical X
gf_point(Y ~ X, data = data_set)
```



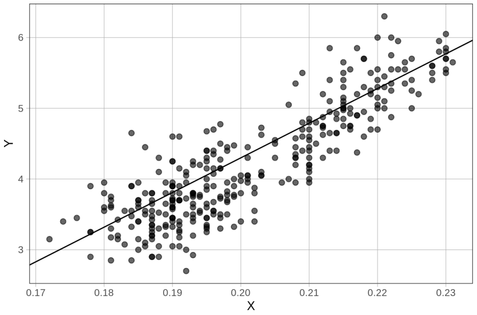
```
# add predictor variables
gf_point(Y ~ X1, data = data_set,
  size = 4, color = ~X2, shape = ~X3)
```



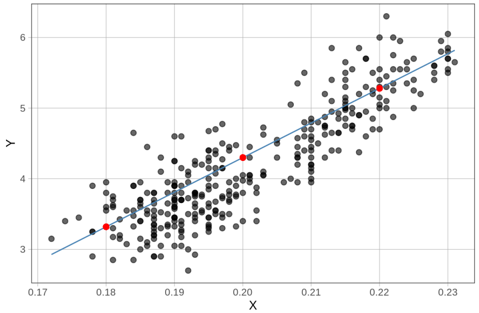
```
# separate facets of scatterplots
gf_point(Y ~ X1, data = data_set) %>%
  gf_facet_wrap(~X3)
```



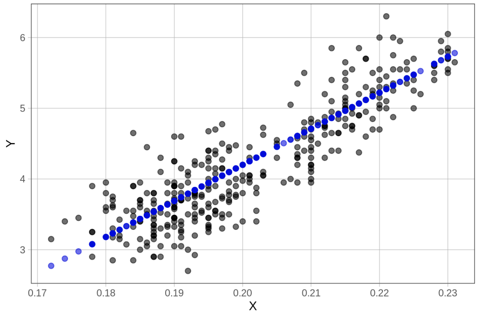
```
# add a line to a scatterplot
gf_point(Y ~ X, data = data_set) %>%
  gf_abline(intercept = -5.5, slope = 49)
```



```
# build custom function
our_function <- function(X){-5.5 + 49*X}

# add function and its predictions to graph
gf_point(Y ~ X, data = data_set) %>%
  gf_function(our_function) %>%
  gf_point(our_function(0.18) ~ 0.18) %>%
  gf_point(our_function(0.20) ~ 0.20) %>%
  gf_point(our_function(0.22) ~ 0.22)
```



```
# add function predictions for all data
gf_point(Y ~ X, data = data_set) %>%
  gf_point(our_function(X) ~ X)
```


Y
3
4
5
6
X
0.17
0.18
0.19
0.20
0.21
0.22
0.23