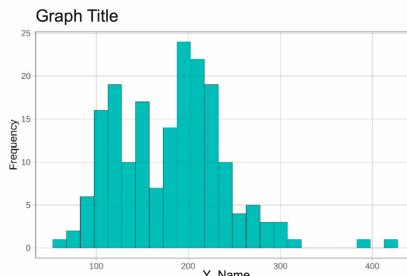


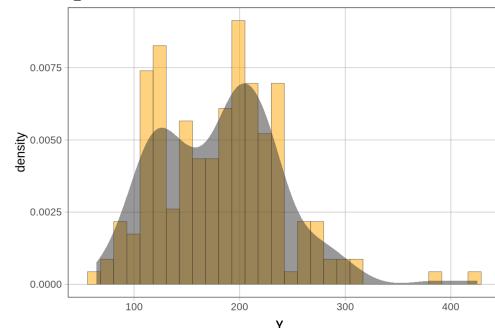
Word Equations	Summary Tables	Simple Statistics
<pre>outcome = explanatory + other stuff Y = X + other stuff</pre>	<pre># compute five-number summary favstats(~ Y, data = data_set) # create frequency table tally(data_set\$Y) tally(~ Y, data = data_set) # tally by condition tally(~ Y < 1900, data = data_set) # two-way frequency table tally(Y ~ X, data = data_set, margin = TRUE, format = "proportion")</pre>	<pre>mean(data_set\$Y) var(data_set\$Y) sd(data_set\$Y) cohensD(Y ~ X, data = data_set) cor(Y ~ X, data = data_set) b1(Y ~ X, data = data_set) b1(my_model) pre(Y ~ X, data = data_set) f(Y ~ X, data = data_set)</pre>
<h3>Basics</h3> <pre>print("Hello world!") # assign value to object my_number <- 5 # combine values into vector my_vector <- c(1, 2, 3) # first element in vector my_vector[1] # orders values or cases sort(my_vector) # arithmetic operations sum(1, 2, 100), +, -, *, / sqrt(157) abs(data_set\$Y) # logical operations >, <, >=, <=, ==, !=, , & # results in a variable with values # of TRUE or FALSE data_set\$C <- data_set\$A > data_set\$B</pre>	<h3>Probability Distribution</h3> <pre># calculate the probability area xpnorm(65.1, data_set\$mean, data_set\$sd) zscore(data_set\$Y)</pre>	<h3>Simulation</h3> <pre># sample without replacement sample(data_set, 6) # sample with replacement resample(data_set, 10) # shuffle the outcome in a graph gf_point(shuffle(Y) ~ X, data = data_set)</pre>
<h3>Data Frame</h3> <pre># structure of data frame str(data_set) # view first/last six rows head(data_set) tail(data_set) # select multiple variables select(data_set, Y1, Y2) # first six rows of selected variables head(select(data_set, Y1, Y2)) # select variable (a column) data_set\$Y # find rows that meet condition data_set[data_set\$Y > 40] filter(data_set, Y > 300) filter(data_set, is.na(Y) == FALSE)</pre>	<pre># arrange rows by variable arrange(data_set, Y) # creates data frame from csv file data_set <- read.csv("file_name", header = TRUE) # convert quantitative variable # to categorical factor(data_set\$Y) factor(data_set\$Y, levels = c(1,2), labels = c("A", "B")) # transform values recode(data_set\$Y, "0" = 0, "1" = 50, "2" = 100) # creates two equal sized groups ntile(data_set\$Y, 2) # convert categorical variable # to quantitative as.numeric(data_set\$Y)</pre>	<h3>Fitting & Evaluating Models</h3> <pre># empty model empty_model <- lm(Y ~ NULL, data = data_set) # use one explanatory variable my_model <- lm(Y ~ X, data = data_set) # model predictions and residuals data_set\$empty_predict <- predict(empty_model) data_set\$empty_resid <- resid(empty_model) # produce ANOVA table anova(empty_model) supernova(my_model)</pre>

Visualizations

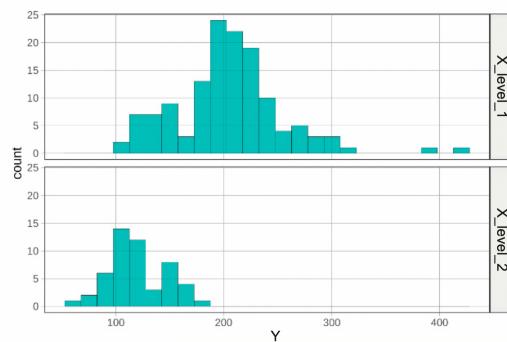
```
gf_histogram(~ Y, data = data_set) %>%
  # change labels
  gf_labs(title = "Graph Title",
  x = "Y_Name", y = "Frequency")
```



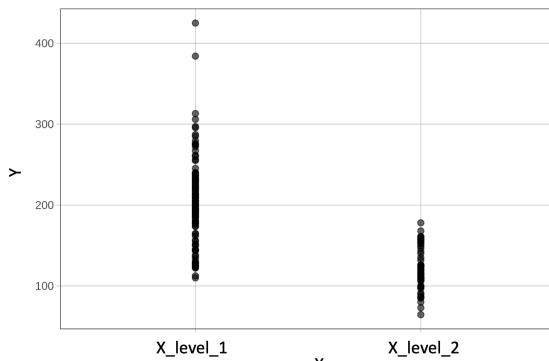
```
gf_dhistogram(~ Y, data = data_set, fill =
"orange") %>%
  gf_density()
```



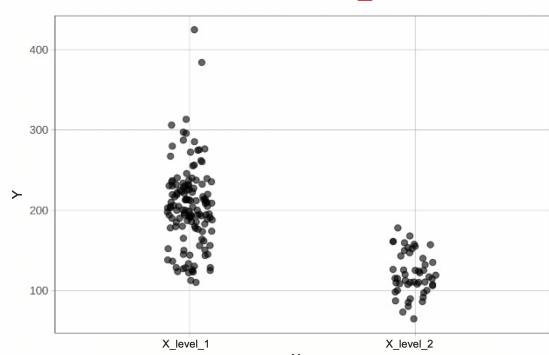
```
# faceted grid of histograms
gf_histogram(~ Y, data = data_set) %>%
  gf_facet_grid(x ~ .)
```



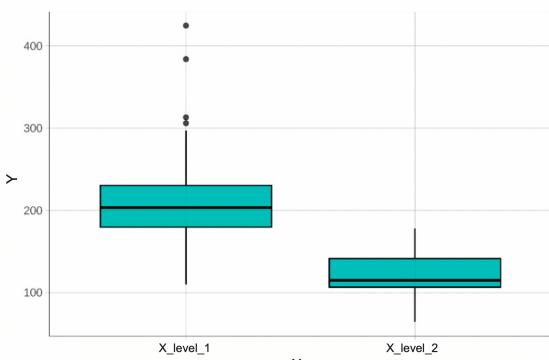
```
gf_point(Y ~ X, data = data_set)
```



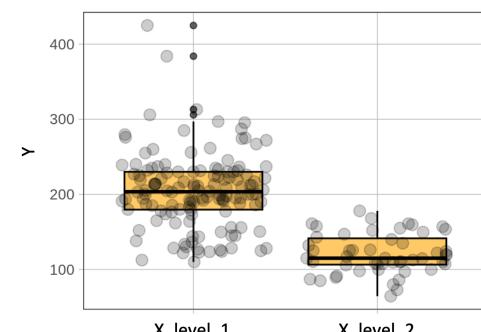
```
gf_jitter(Y ~ X, data = data_set)
```



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



```
gf_boxplot(Y ~ X, data = data_set, fill =
"orange") %>%
  gf_jitter(height = 0, alpha = .2, size = 3)
```



```
gf_bar( ~ Y, data = data_set)
```



```
gf_point(Y ~ X, data = data_set) %>%
  # add model predictions as red points
  gf_point(Y ~ X, shape = 1, size = 3,
  color = "firebrick") %>%
  # add best fitting model as a red line
  gf_model(my_model, color = "red")
```

