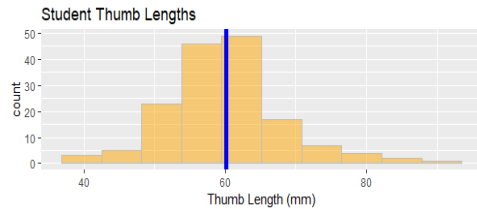


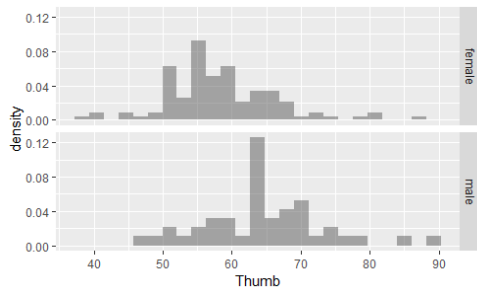
Basics <pre>print("hello") # assigns value to object my_number <- 5 # combines elements into vector my_vector <- c(1,2,3) # first element in vector my_vector[1] # variable in data frame Fingers\$Sex</pre>	Tables <pre>tally(my_vector) tally(~ Condition, data = MindsetMatters) tally(~ Thumb > 65, data = Fingers) tally(Thumb ~ Sex, data = Fingers, margins = TRUE, format = "proportion")</pre>	Fitting and Evaluating Models <pre>empty_model <- lm(Thumb ~ NULL, data = Fingers) Sex_model <- lm(Thumb ~ Sex, data = Fingers) predict(empty_model) resid(empty_model) anova(empty_model) supernova(Sex_model) t.test(Tip ~ Condition, data = TipExperiment, var.equal=TRUE) pairwise(game_model, correction="none")</pre>	Data <pre>str(MindsetMatters) head(MindsetMatters) tail(MindsetMatters) sort(my_vector) arrange(Fingers, Thumb) # selects variables select(Fingers, Sex, RaceEthnic, Thumb) # selects cases filter(Fingers, SLast != "NA") head(select(Fingers, Thumb))</pre>
Operators <pre>sum(1,2,100) +, -, *, / >, <, >=, <=, ==, != # results in TRUE or FALSE Fingers\$RingLonger <- Fingers\$Ring > Fingers\$Index abs(Fingers\$Residual) Fingers\$Residual^2 sqrt(157)</pre>	Simple Statistics <pre>mean(Fingers\$Thumb) var(Fingers\$Thumb) sd(Fingers\$Thumb) favstats(~ Wt, data = MindsetMatters) cohensD(Thumb ~ Sex, data = Fingers) cor(Thumb ~ Height, data = Fingers) b1(Thumb ~ Sex, data = Fingers) b1(Sex_model) # PRE and fVal work like b1 PRE(Sex_model) fVal(Sex_model)</pre>	Probability Distributions <pre>xpnorm(65.1, Thumb_stats\$mean, Thumb_stats\$sd) zscore(Fingers\$Thumb) # returns t at this probability qt(.975, df = 999) # returns F at this probability qf(.95, df1 = 1, df2 = 100) # CI using t dist. confint(empty_model)</pre>	<pre>as.numeric(Fingers\$Interest) factor(Fingers\$Sex) factor(Fingers\$Sex, levels = c(1,2), labels = c("female", "male")) recode(Fingers\$Job, "0" = 0, "1" = 50, "2" = 100) # creates two equal sized groups ntile(Fingers\$Height, 2) # creates data frame from csv file new_dataframe <- read.csv("long- csv-link-from-published-google- spreadsheet", header = TRUE)</pre>
Simulation & Resampling <pre># sample without replacement sample(Fingers\$Thumb, 10) # sample with replacement resample(Fingers\$Thumb, 157) do(3) * resample (Fingers\$Thumb,10) # mixes up values in a variable shuffle(servers\$random_groups_1) # simulates sampling 10000 Thumbs from a normal dist. sim_Thumb <- rnorm(10000, Thumb_stats\$mean, Thumb_stats\$sd)</pre>	<pre># puts simulated Thumbs into data frame sim_Pop <- data.frame(sim_Thumb) # simulates sampling dist.of means sim_SDoM <- do(10000) * mean(rnorm(157, Thumb_stats\$mean, Thumb_stats\$sd)) # bootstraps sampling dist. of means bootSDoM <- do(10000) * mean(resample(Fingers\$Thumb,157)) # bootstraps sampling dist. of b1s, centered on sample b1 SDob1 <- do(10000) * b1(Tip ~ Condition, data = resample(TipExperiment, 44))</pre>	<pre># randomizes sampling dist. of b1s, centered on 0 SDob1 <- do(10000) * b1(Tip ~ shuffle(Condition), data = TipExperiment) # get the middle 95 percent of the distribution middle(b1, .95) # randomizes sampling dist. of PREs SDoPRE <- do(10000) * PRE(Tip ~ shuffle(Condition), data = TipExperiment) # randomizes sampling dist. of Fs sdoF <- do(10000) * fVal(Tip ~ shuffle(Condition), data = TipExperiment)</pre>	<pre># plots sampling dist. gf_histogram(~ fVal, data = SDoF, fill = ~fVal>sample_F) # counts extreme Fs tally(~fVal>sample_F, data = SDoF) # count the number of b1s at the upper and lower extreme tally(sdob1\$b1 > sample_b1 sdob1\$b1 < -sample_b1) # fill the lower .95 of this histogram with a different color gf_histogram(~ fVal, data = sdoF, fill = ~lower(fVal, .95)) # calculate a p-value using the F- distribution xpf(sample_F, df1 = , df2 =)</pre>

Plots

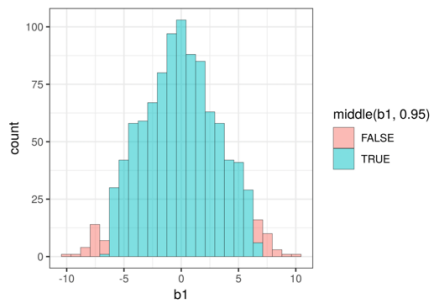
```
gf_histogram(~ Thumb, data = Fingers, fill =
"orange", color = "gray", bins = 10) %>%
# changes labels
gf_labs(title= "Student Thumb Lengths", x =
"Thumb Length (mm)") %>%
# adds density curve to a histogram
gf_density() %>%
# adds vertical line
gf_vline(xintercept = ~mean, data =
Thumb_stats, color = "blue", size = 2)
```



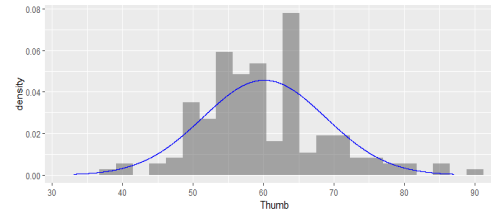
```
gf_dhistogram(~ Thumb, data = Fingers) %>%
gf_facet_grid(Sex ~ .)
```



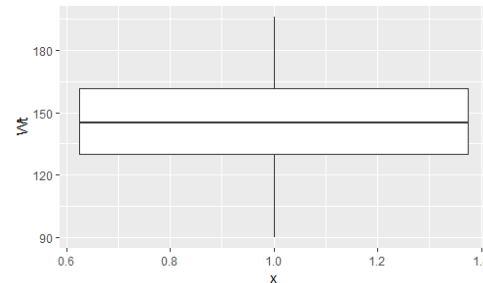
```
gf_histogram(~b1, data = sdob1, fill =
~middle(b1, .95))
```



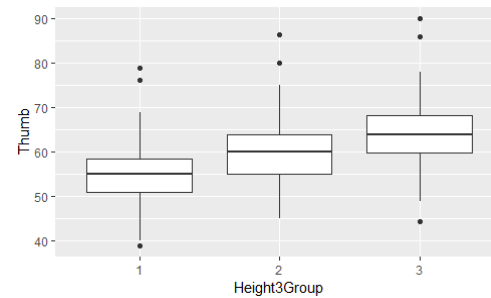
```
gf_dist("norm", color = "blue", params =
list(Thumb_stats$mean, Thumb_stats$sd))
```



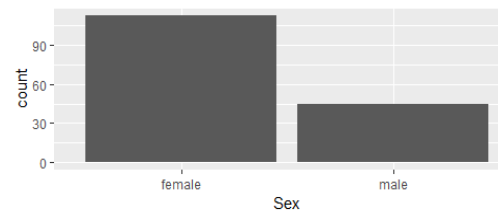
```
gf_boxplot(Wt ~ 1, data = MindsetMatters)
```



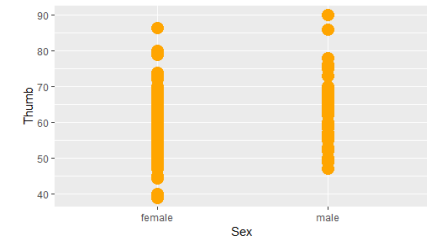
```
gf_boxplot(Thumb ~ Height3Group, data =
Fingers)
```



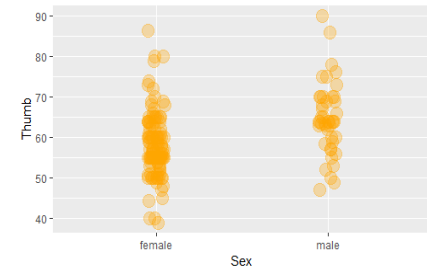
```
# creates bar graph
gf_bar(~ Sex, data = Fingers)
```



```
gf_point(Thumb ~ Sex, data = Fingers, color =
"orange", size = 5)
```

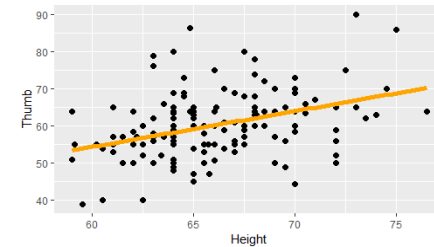


```
gf_jitter(Thumb ~ Sex, data = Fingers, color =
"orange", size = 5, alpha = .5)
```



```
gf_point(Thumb ~ Height, data = Fingers, size
= 2) %>%
```

```
# adds a regression line
gf_lm(color = "orange", size = 2)
```



```
pairwise(game_model, plot = TRUE)
```

