Basics

```
print("hello")
# assigns value to object
mynumber <- 5
# combines elements into vector
myvector <- c(1,2,3)
# first element in vector
myvector[1]
# variable in data frame
Fingers$Sex
```

Tables

```
tally(myvector)
tally(~ Condition, data =
MindsetMatters)
tally(~ Thumb > 65, data = Fingers)
tally(Thumb ~ Sex, data = Fingers,
margins = TRUE, format =
"proportion")
```

Fitting and Evaluating Models

```
Empty.model <- lm(Thumb ~ NULL,
data = Fingers)
Sex.model <- lm(Thumb ~ Sex, data
= Fingers)
Sex.fun<-makeFun(Sex.model)
Sex.fun("male")

predict(Empty.model)

resid(Empty.model)
anova(Empty.model)
supernova(Sex.model)
```

Data

```
str(MindsetMatters)
head(MindsetMatters)
tail(MindsetMatters)
sort(myvector)
arrange(Fingers, Thumb)

# selects variables
select(Fingers, Sex, RaceEthnic,
Thumb)
# selects cases
filter(Fingers, SSLast != "NA")
head(select(Fingers, Thumb))

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)

aggregate(Happiness ~ Region, data
= HappyPlanetIndex, FUN = mean)
```

Operators

```
sum(1,2,100)
+, -, *, /
>, <, >=, <=, ==, !=

# results in TRUE or FALSE
Fingers$RingLonger <-
Fingers$Ring > Fingers$Index

abs(Fingers$Residual)
Fingers$Residual^2
sqrt(157)
```

Simple Statistics

```
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)
```

Probability Distributions

```
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)
```

Simulation & Resampling

```
# 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$RandomGroups1)
```

```
# simulates sampling 10000 Thumbs
from a normal dist.
simThumb <- rnorm(10000,
Thumb.stats$mean, Thumb.stats$sd)
# puts simulated Thumbs into data
frame
simPop <- data.frame(simThumb)

# simulates sampling dist.of means
simSDoM <- do(10000) *
mean(rnorm(157, Thumb.stats$mean,
Thumb.stats$sd))
# bootstraps sampling dist. of means
bootSDoM <- do(10000) *
mean(resample(Fingers$Thumb,157))
```

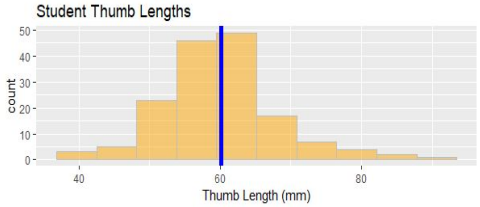
```
# bootstraps sampling dist. of
b1s, centered on sample b1
SDob1 <- do(10000) * b1(Tip ~
Condition, data =
resample(TipExperiment, 44))

# randomizes sampling dist. of
b1s, centered on 0
SDob1 <- do(10000) * b1(Tip ~
shuffle(Condition), data =
TipExperiment)
```

```
# 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)

# plots sampling dist.
gf_histogram(~ fVal, data = SDoF,
fill = ~fVal>sampleF)
# counts extreme Fs
tally(~fVal>sampleF, data = SDoF)
```

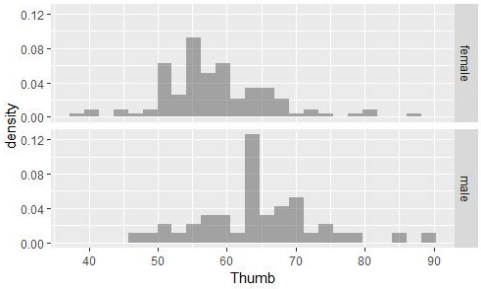
Version 2.2

Plots

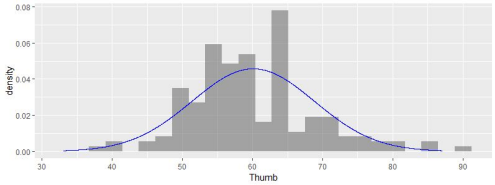
```
gf_histogram( ~ Thumb, data = Fingers, fill = "orange", color = "gray", bins = 10) %>%
  # changes labels
  gf_labs(title= "Student Thumb Lengths", x = "Thumb Length (mm)") %>%
  # adds density curve to a histogram
  gf_density() %>%
  # adds vertical line
  gf_vline(xintercept = ~mean, data = Thumb.stats,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



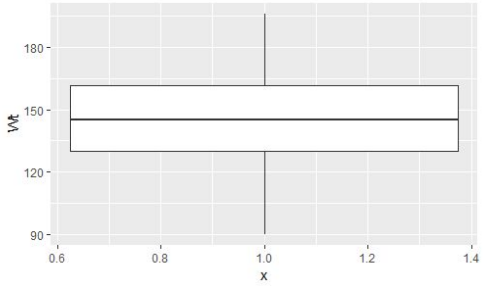

```
gf_dhistogram(~ Thumb, data = Fingers) %>%
  gf_facet_grid(Sex ~ .)
```



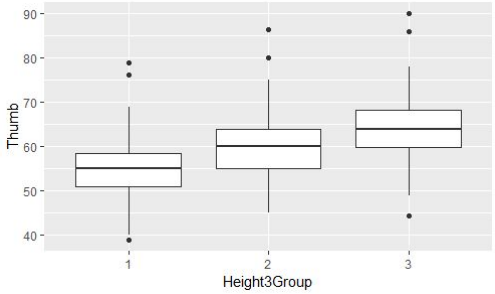

```
gf_dist("norm", color = "blue", params = list(Thumb.stats$mean, Thumb.stats$sd))
```



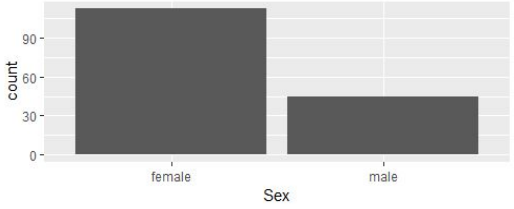

```
gf_boxplot(Wt ~ 1, data = MindsetMatters)
```



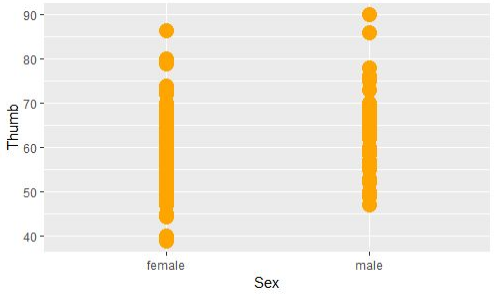

```
gf_boxplot(Thumb ~ Height3Group, data = Fingers)
```



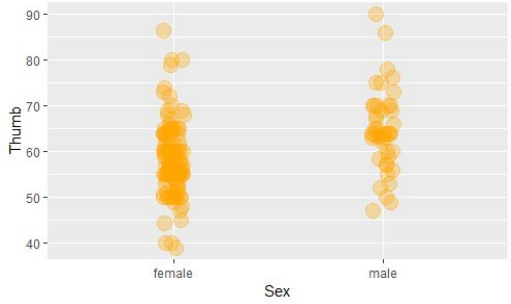

```
# creates bar graph
gf_bar( ~ Sex, data = Fingers)
```



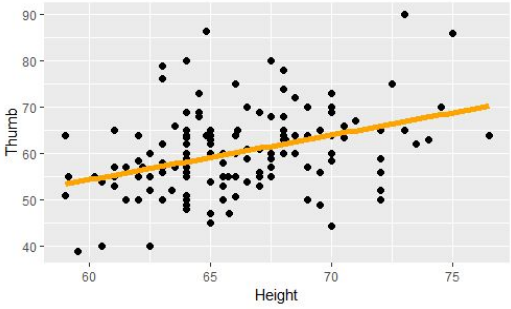

```
gf_point(Thumb ~ Sex, data = Fingers, color = "orange", size = 5)
```




```
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)
```



Version 2.2