Help

```
?command
help(commandname)
```

Basic Calculations

Basic calculation works like a calculator.

```
# basic ops: + - * / ^ ( )
log(); exp(); sqrt()
```

Formula Interface

The following syntax (often with some parts omitted) is used for graphical summaries, numerical summaries, and inference procedures.

```
goal(y ~ x | z, data = mydata, ...)
```

For plots:

- `y`: is y-axis variable
- `x`: is x-axis variable
- `z`: conditioning variable (separate panels)

For other things:

'`y ~ x | z`' can usually be read '`y` is modeled by (or depends on) `x` differently for each `z`'.

See the sampler for examples.

Categorical Variable Description

```
tally()
gf_bar()
gf_props()
```

Numerical Summaries

These functions have a formula interface to match plotting.

```
favstats()
mean()
median()
sd()
quantile()
IQR()
```

Graphics

```
gf_bar()
gf_histogram()
gf_dotplot()
gf_boxplot()
gf_point()
gf_smooth()   # End previous gf_point()
              # line with a  |>  sign
annotate()    # End previous line with a + sign
```

Correlation and Regression

```
cor()
model <- lm()
summary(model)
predict(model)
resid(model)
```

Data

```
nrow()
dim()
names()
head()
tail()
View()
```

Special Commands

```
c()
 |> # Pipe
+   # Continuation

filter()                # Subset rows
slice()                 # Select specific rows
mydata[-c(3,23,36), ]   # Remove specified rows
mydata[, c(2,5)]        # Show specified columns
mutate()                # Create new variables
```

Here are a few examples. These are not exhaustive but should be seen as representative of the kinds of things we will be doing.

One Categorical

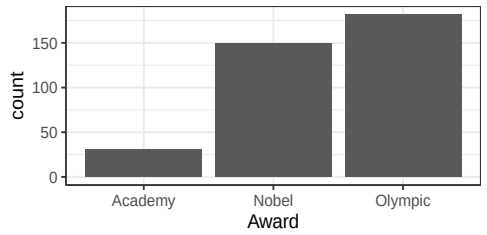
```
tally(~ Award, data = StudentSurvey)

Award
Academy   Nobel Olympic
     31     149     182
```

```
tally(~ Award, data = StudentSurvey,
      format = "percent")

Award
Academy   Nobel Olympic
  8.564  41.160  50.276
```

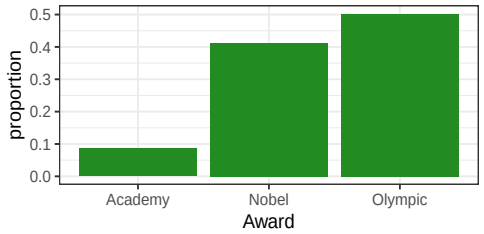
```
tally(~ Award, data = StudentSurvey,
      format = "proportion")

Award
Academy   Nobel Olympic
0.08564 0.41160 0.50276
```

```
tally(~ Award, data = StudentSurvey,
      format = "proportion", margins = TRUE)

Award
Academy   Nobel Olympic   Total
0.08564 0.41160 0.50276 1.00000
```

```
gf_bar(~Award,data=StudentSurvey)
```




```
gf_props(~Award,data=StudentSurvey,
         fill = "forestgreen")
```



Two Categorical

```
tally(~ Award + Smoke , data = StudentSurvey)

         Smoke
Award      No Yes
  Academy  29   2
  Nobel   129  20
  Olympic 161  21
```

```
tally(~ Award + Smoke , format = "proportion",
      data = StudentSurvey)

         Smoke
Award           No      Yes
  Academy 0.080110 0.005525
  Nobel   0.356354 0.055249
  Olympic 0.444751 0.058011
```

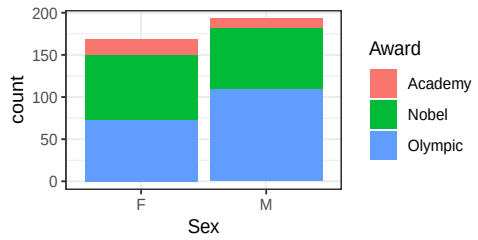
```
tally(~ Award + Smoke , format = "percent",
      margins = TRUE, data = StudentSurvey)

         Smoke
Award           No      Yes    Total
  Academy   8.0110   0.5525   8.5635
  Nobel    35.6354   5.5249  41.1602
  Olympic  44.4751   5.8011  50.2762
  Total    88.1215  11.8785 100.0000
```

```
tally(~ Award | Smoke , data=StudentSurvey,
      format = "proportion")

         Smoke
Award          No     Yes
  Academy 0.09091 0.04651
  Nobel   0.40439 0.46512
  Olympic 0.50470 0.48837
```

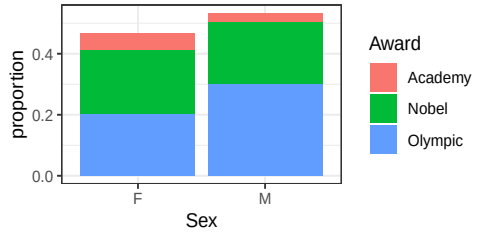
```
tally(~ Award | Smoke , data=StudentSurvey,
      format = "percent")

         Smoke
Award         No    Yes
  Academy  9.091  4.651
  Nobel   40.439 46.512
  Olympic 50.470 48.837
```

```
tally(~ Award | Smoke, data=StudentSurvey,
      format = "proportion", margins = TRUE)

         Smoke
Award          No     Yes
  Academy 0.09091 0.04651
  Nobel   0.40439 0.46512
  Olympic 0.50470 0.48837
  Total   1.00000 1.00000
```

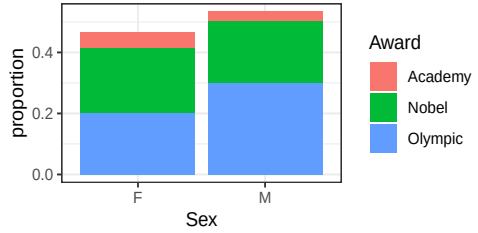
```
gf_bar(~Sex, fill = ~Award, data=StudentSurvey)
```



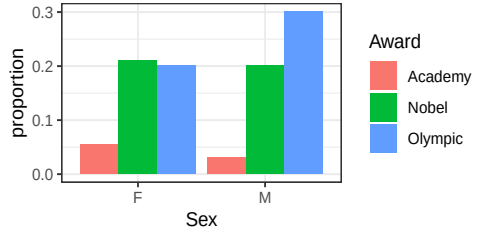

```
gf_props(~Sex, fill = ~Award, data=StudentSurvey)
```



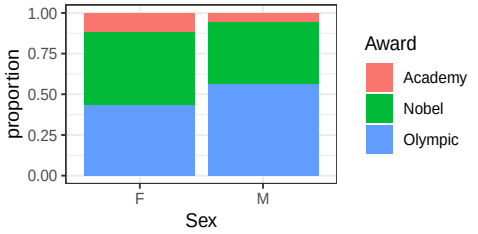

```
gf_props(~Sex, fill = ~Award, position = "stack",
         data=StudentSurvey)
```




```
gf_props(~Sex, fill = ~Award,
         position = "dodge", data=StudentSurvey)
```

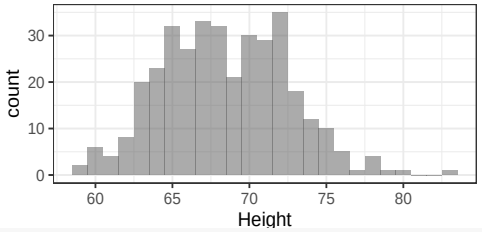



```
gf_props(~Sex, fill = ~Award,
         position = "fill", data=StudentSurvey)
```

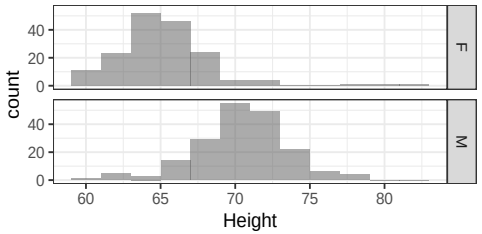


One Quantitative

```
gf_histogram(~ Height, data = StudentSurvey)
```



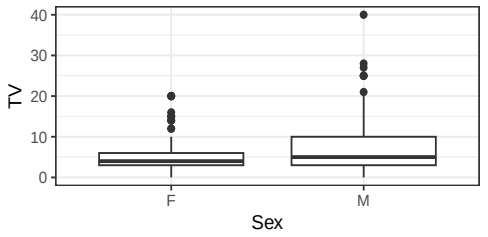

```
gf_histogram(~ Height | Sex ~ .,
             data = StudentSurvey, binwidth = 2)
```



One Quantitative and One Categorical

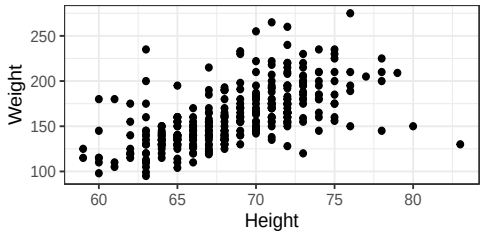
```
favstats(TV ~ Sex, data = StudentSurvey)

  Sex min Q1 median Q3 max  mean    sd
1   F   0  3      4  6  20 5.237 4.100
2   M   0  3      5 10  40 7.620 6.427
   n missing
1 169       0
2 192       1
```

```
gf_boxplot(TV ~ Sex, data = StudentSurvey)
```

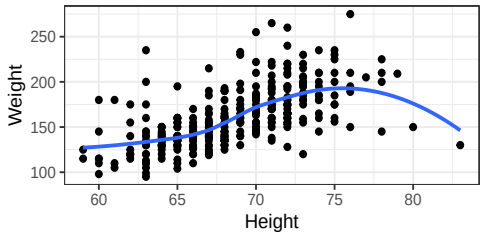


Two Quantitative

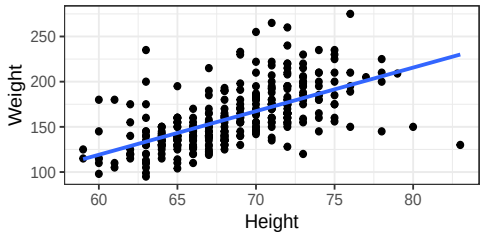
```
gf_point(Weight ~ Height, data = StudentSurvey)
```



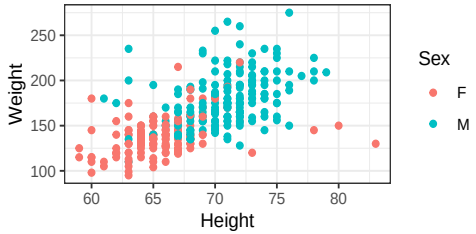

```
gf_point(Weight ~ Height, data = StudentSurvey)  |>
  gf_smooth()
```



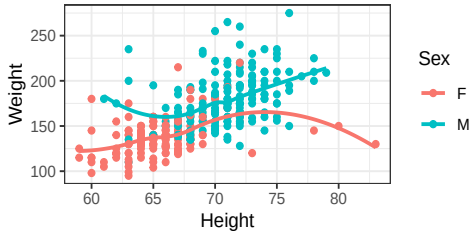

```
gf_point(Weight ~ Height, data = StudentSurvey)  |>
  gf_smooth(method = "lm")
```



```
gf_point(Weight ~ Height, color = ~ Sex, data = StudentSurvey)
```




```
gf_point(Weight ~ Height, color = ~ Sex, data = StudentSurvey)  |>
  gf_smooth()
```




```
model <- lm(Weight ~ Height, data = StudentSurvey)
summary(model)


Call:
lm(formula = Weight ~ Height, data = StudentSurvey)

Residuals:
    Min      1Q  Median      3Q     Max
-100.05  -13.81   -3.06   11.76  101.41

Coefficients:
            Estimate Std. Error t value
(Intercept) -170.269     22.412    -7.6
Height         4.823      0.327    14.8
            Pr(>|t|)
(Intercept)  2.8e-13
Height       < 2e-16

Residual standard error: 24.9 on 350 degrees of freedom
  (10 observations deleted due to missingness)
Multiple R-squared:  0.384,Adjusted R-squared:  0.382
F-statistic:  218 on 1 and 350 DF,  p-value: <2e-16
```