Chapter 19

Conducting Research

Presenting Tables and Graphs

Doing Economics: A Primer

Posing a question

- Topic – the beginning of your search, not the end
  - Narrow it down
  - Talk to (other) faculty
  - Research existing literature
  - Find a niche or replicate?
- Assess feasibility
  - Are data sufficient?
  - Are my tools and other resources sufficient?
- Focus on a testable hypothesis

Literature Review

- Review the *literature*, not individual contributions
  - Organize papers into different "threads" of the literature -Differentiate by type of analysis (theory vs. empirical, causal vs. descriptive), type of result (positive, negative, none), type of data (geographic location, macro vs. micro, time periods), sub-topics...
- State your paper's contribution
  - Niche
  - Replication
  - ~~Literature Review~~
  - ~~Meta-Analysis~~
- Databases

Research Databases

- Library: EBSCO Host, JSTOR, Science Direct
- Google Scholar
  - Pro tip: Select "Endnote" as your bibliography manager in "Search results" and search and add "SAU" to the "Library links" in settings.

Structure of the Paper

- Introduction
- Conceptual Framework
- Data and Methods
- Results and Discussion
- Conclusion

Data Collection

- Online Sources
  - Formatted Electronic Data (.xls, .csv, .dta, .dat, etc. . . .)
  - PDFs (require some manual entry/editing)
- Merging Data
  - ID variable (Key)
  - Dealing with mismatches?
- Cleaning Data
  - Dealing with (and reformatting) NA's
  - Naming, labelling, and storing data (.Rdata files, .Rproj files, and project directories)

Summarizing Data

Single Variables

Do the following: - Explore the objects in your environment using *ls()* - Using the *ceosal1* data, separately calculate the mean, median, and standard deviation of the *salary* variable. - Calculate the summary statistics for the *salary* variable.

```
ls()
```

```
## [1] "affairs" "ceosal1"
```

```
mean(ceosal1$salary)
```

```
## [1] 1281.12
```

```
median(ceosal1$salary)
```

```
## [1] 1039
```

```
sd(ceosal1$salary)
```

```
## [1] 1372.345
```

```
summary(ceosal1$salary)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     223     736    1039    1281    1407   14822
```

Summarizing an Entire Data Frame

- Calculate the summary statistics for all of the variables in *ceosal1*.

```
summary(ceosal1)
```

```
##      salary         pcsalary          sales              roe
##  Min.   :  223   Min.   :-61.00   Min.   :  175.2   Min.   : 0.50
##  1st Qu.:  736   1st Qu.: -1.00   1st Qu.: 2210.3   1st Qu.:12.40
##  Median : 1039   Median :  9.00   Median : 3705.2   Median :15.50
##  Mean   : 1281   Mean   : 13.28   Mean   : 6923.8   Mean   :17.18
##  3rd Qu.: 1407   3rd Qu.: 20.00   3rd Qu.: 7177.0   3rd Qu.:20.00
##  Max.   :14822   Max.   :212.00   Max.   :97649.9   Max.   :56.30
##      pcroe             ros             indus           finance
##  Min.   :-98.9   Min.   :-58.0   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:-21.2   1st Qu.: 21.0   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : -3.0   Median : 52.0   Median :0.0000   Median :0.0000
```

```
##  Mean   : 10.8   Mean   : 61.8   Mean   :0.3206   Mean   :0.2201
##  3rd Qu.: 19.5   3rd Qu.: 81.0   3rd Qu.:1.0000   3rd Qu.:0.0000
##  Max.   :977.0   Max.   :418.0   Max.   :1.0000   Max.   :1.0000
##     consprod         utility          lsalary          lsales
##  Min.   :0.0000   Min.   :0.0000   Min.   :5.407   Min.   : 5.166
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:6.601   1st Qu.: 7.701
##  Median :0.0000   Median :0.0000   Median :6.946   Median : 8.217
##  Mean   :0.2871   Mean   :0.1722   Mean   :6.950   Mean   : 8.292
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:7.249   3rd Qu.: 8.879
##  Max.   :1.0000   Max.   :1.0000   Max.   :9.604   Max.   :11.489
```

Creating Pretty Tables

Standard Output

Create summary tables of *ceosal1* using the *tbl_summary* function in the *gtsummary* package.

```
tbl_summary(ceosal1)
```

Characteristic	N = 209
salary	1,039 (736, 1,407)
pcsalary	9 (-1, 20)
sales	3,705 (2,210, 7,177)
roe	16 (12, 20)
pcroe	-3 (-21, 20)
ros	52 (21, 81)
indus	67 (32%)
finance	46 (22%)
consprod	60 (29%)
utility	36 (17%)
lsalary	6.95 (6.60, 7.25)
lsales	8.22 (7.70, 8.88)

Means and Standard Deviations

Input the request from the feedback below.

```
tbl_summary(ceosal1, statistic = list(all_continuous() ~ "{mean} ({sd})"), digits = list(all_continuous() ~ c(2,2)))
```

Characteristic	N = 209
salary	1,281.12 (1,372.35)
pcsalary	13.28 (32.63)
sales	6,923.79 (10,633.27)
roe	17.18 (8.52)
pcroe	10.80 (97.22)
ros	61.80 (68.18)
indus	67 (32%)
finance	46 (22%)
consprod	60 (29%)
utility	36 (17%)

Characteristic	**N = 209**
lsalary	6.95 (0.57)
lsales	8.29 (1.01)

Relationships between Variables

Correlation Matrix

Calculate the following:

- The correlation of salary with ROE;
- The correlation matrix for all numeric variables.

```
cor(ceosal1$salary, ceosal1$roe)
```

```
## [1] 0.1148417
```

```
cor(ceosal1)
```

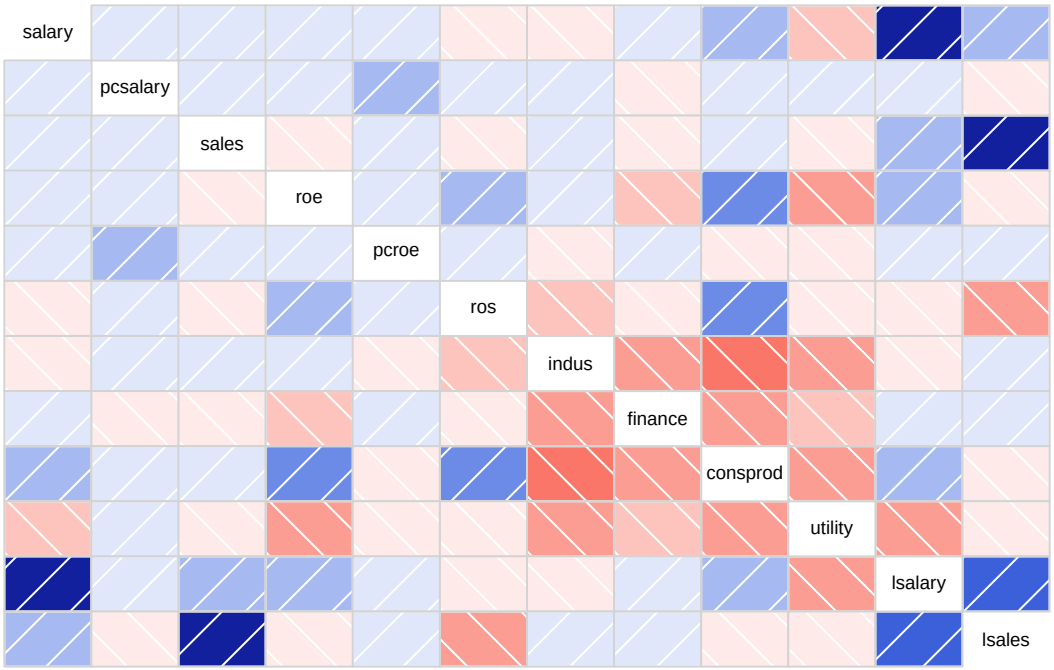
```
##                salary     pcsalary        sales          roe        pcroe
## salary    1.000000000  0.008672195  0.119869489  0.114841735  0.028710443
## pcsalary  0.008672195  1.000000000  0.017010310  0.087335193  0.207962311
## sales     0.119869489  0.017010310  1.000000000 -0.055385713  0.005594043
## roe       0.114841735  0.087335193 -0.055385713  1.000000000  0.004191102
## pcroe     0.028710443  0.207962311  0.005594043  0.004191102  1.000000000
## ros      -0.033681897  0.137778486 -0.136087621  0.274918807  0.128939567
## indus    -0.071133642  0.004435904  0.093608307  0.013461034 -0.029601568
## finance   0.024753643 -0.090806118 -0.054073079 -0.178530972  0.091719844
## consprod  0.204546281  0.051998671  0.069174857  0.408517256 -0.015684306
## utility  -0.184309255  0.031852925 -0.139244839 -0.310193625 -0.045259914
## lsalary   0.794208152  0.043842774  0.281285760  0.208499203  0.107694007
## lsales    0.194092092 -0.065224367  0.742921432 -0.122553150  0.023288281
##                  ros        indus     finance    consprod     utility
## salary   -0.03368190 -0.071133642  0.02475364  0.20454628 -0.18430926
## pcsalary  0.13777849  0.004435904 -0.09080612  0.05199867  0.03185292
## sales    -0.13608762  0.093608307 -0.05407308  0.06917486 -0.13924484
## roe       0.27491881  0.013461034 -0.17853097  0.40851726 -0.31019363
## pcroe     0.12893957 -0.029601568  0.09171984 -0.01568431 -0.04525991
## ros       1.00000000 -0.209503670 -0.10798696  0.34782108 -0.03929917
## indus    -0.20950367  1.000000000 -0.36490376 -0.43588881 -0.31334403
## finance  -0.10798696 -0.364903764  1.00000000 -0.33710687 -0.24233342
## consprod  0.34782108 -0.435888810 -0.33710687  1.00000000 -0.28947475
## utility  -0.03929917 -0.313344032 -0.24233342 -0.28947475  1.00000000
## lsalary  -0.07456453 -0.016145538  0.10084137  0.22028456 -0.35461477
## lsales   -0.35032965  0.060277561  0.03902509 -0.01907079 -0.09447193
##              lsalary      lsales
```

```
## salary    0.79420815  0.19409209
## pcsalary  0.04384277 -0.06522437
## sales     0.28128576  0.74292143
## roe       0.20849920 -0.12255315
## pcroe     0.10769401  0.02328828
## ros      -0.07456453 -0.35032965
## indus    -0.01614554  0.06027756
## finance   0.10084137  0.03902509
## consprod  0.22028456 -0.01907079
## utility  -0.35461477 -0.09447193
## lsalary   1.00000000  0.45914817
## lsales    0.45914817  1.00000000
```

Pretty Correlation Matrix

Use the *corrgram* function (part of the *corrgram* package) to create a correlogram.

```
corrgram(ceosal1)
```



Pretty *Numerical* Correlation Matrix

Input the suggestion from the feedback below.

```
corrgram(ceosal1, panel = panel.cor)
```


salary 0.01 0.12 0.11 0.03 −0.03 −0.07 0.02 0.20 −0.18 0.79 0.19
0.01 pcsalary 0.02 0.09 0.21 0.14 0.00 −0.09 0.05 0.03 0.04 −0.07
0.12 0.02 sales −0.06 0.01 −0.14 0.09 −0.05 0.07 −0.14 0.28 0.74
0.11 0.09 −0.06 roe 0.00 0.27 0.01 −0.18 0.41 −0.31 0.21 −0.12
0.03 0.21 0.01 0.00 pcroe 0.13 −0.03 0.09 −0.02 −0.05 0.11 0.02
−0.03 0.14 −0.14 0.27 0.13 ros −0.21 −0.11 0.35 −0.04 −0.07 −0.35
−0.07 0.00 0.09 0.01 −0.03 −0.21 indus −0.36 −0.44 −0.31 −0.02 0.06
0.02 −0.09 −0.05 −0.18 0.09 −0.11 −0.36 finance −0.34 −0.24 0.10 0.04
0.20 0.05 0.07 0.41 −0.02 0.35 −0.44 −0.34 consprod −0.29 0.22 −0.02
−0.18 0.03 −0.14 −0.31 −0.05 −0.04 −0.31 −0.24 −0.29 utility −0.35 −0.09
0.79 0.04 0.28 0.21 0.11 −0.07 −0.02 0.10 0.22 −0.35 lsalary 0.46
0.19 −0.07 0.74 −0.12 0.02 −0.35 0.06 0.04 −0.02 −0.09 0.46 lsales

Factor Variables

Crosstabulation Tables

Using the *affairs* data, do the following:

- Generate the factor variables *haskids* and *marriage* for kids and ratemarr labels with labels *no/yes* and *very unhappy/unhappy/average/happy/very happy*, respectively;
- Create a table that displays the proportions of each outcome of marriage happiness rating;
- Create a table that displays the proportions of each outcome of marriage happiness rating *and having kids*.

```
affairs$haskids <- factor(affairs$kids, labels = c("no","yes"))
affairs$marriage <- factor(affairs$ratemarr, labels = c("very unhappy","unhappy","average","happy", "very happy"))
prop.table(table(affairs$marriage))
```

```
##
## very unhappy      unhappy      average        happy   very happy
##    0.0266223    0.1098170    0.1547421    0.3227953    0.3860233
```

```
prop.table(table(affairs$marriage,affairs$haskids))
```

```
##
##                          no         yes
##   very unhappy 0.004991681 0.021630616
##   unhappy      0.013311148 0.096505824
##   average      0.039933444 0.114808652
##   happy        0.066555740 0.256239601
##   very happy   0.159733777 0.226289517
```

Pretty Crosstab Tables

- Replicate the previous tables using "tbl_cross" (using percent to generate percentages instead of proportions) to generate a more attractive layout that you can save as html.
- Combine counts and percentages in one table with each cell displaying percentages in parentheses next to its count.

```
tbl_cross(affairs, row = marriage, col = haskids)
```

	no	yes	Total
marriage			
very unhappy	3	13	16
unhappy	8	58	66
average	24	69	93

	no	yes	Total
happy	40	154	194
very happy	96	136	232
Total	171	430	601

```
tbl_cross(affairs, row = marriage, col = haskids, statistic = "{p}%")
```

	no	yes	Total
marriage			
very unhappy	0.5%	2.2%	2.7%
unhappy	1.3%	9.7%	11%
average	4.0%	11%	15%
happy	6.7%	26%	32%
very happy	16%	23%	39%
Total	28%	72%	100%

```
tbl_cross(affairs, row = marriage, col = haskids, percent = 'cell')
```

	no	yes	Total
marriage			
very unhappy	3 (0.5%)	13 (2.2%)	16 (2.7%)
unhappy	8 (1.3%)	58 (9.7%)	66 (11%)
average	24 (4.0%)	69 (11%)	93 (15%)
happy	40 (6.7%)	154 (26%)	194 (32%)
very happy	96 (16%)	136 (23%)	232 (39%)
Total	171 (28%)	430 (72%)	601 (100%)

Graphs

Base R Graphics

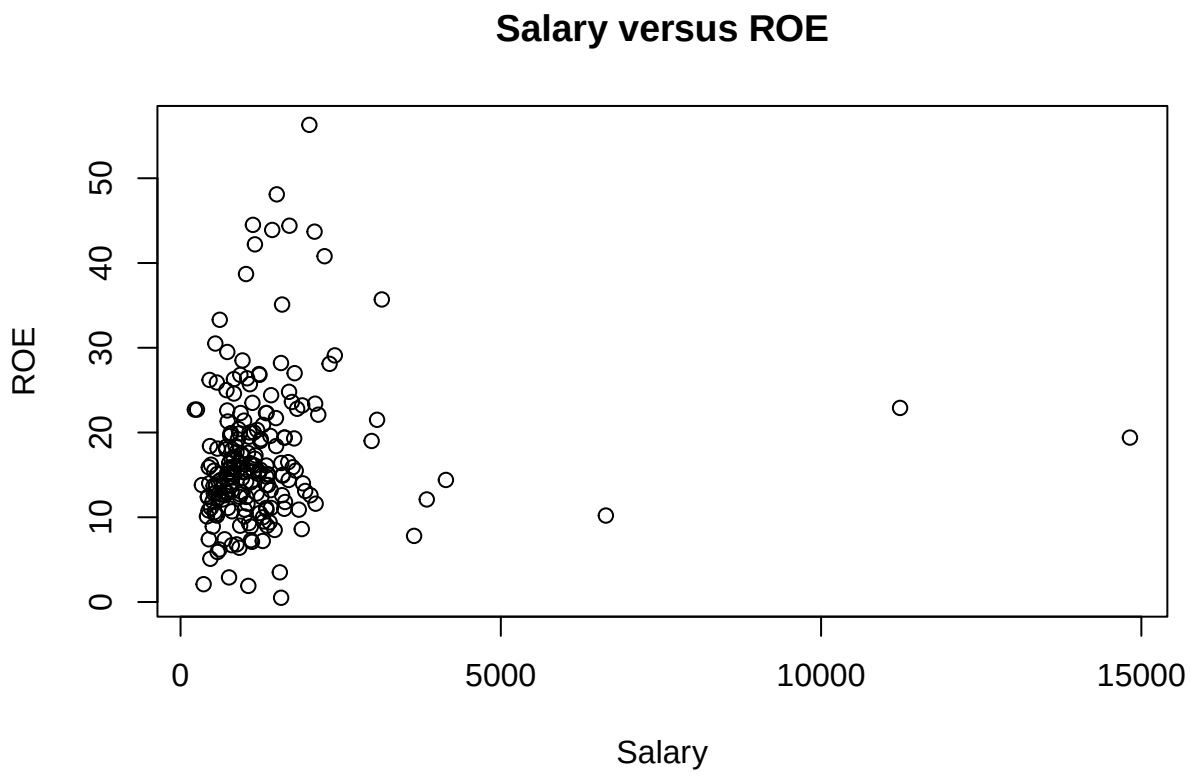
Plot the following:

- A histogram of CEO salary *relative* frequencies using the base-graphics hist() function;
- A scatterplot of CEO salaries with ROE using the plot() function.

Make sure your plots have descriptive (English) titles: *Histogram of Salary*, *Salary versus ROE*, *Salary*, and *ROE*.

```
hist(ceosal1$salary, main = "Histogram of Salary", xlab = "Salary", freq = FALSE)
```

```
plot(ceosal1$salary, ceosal1$roe, main = "Salary versus ROE", xlab = "Salary", ylab = "ROE")
```

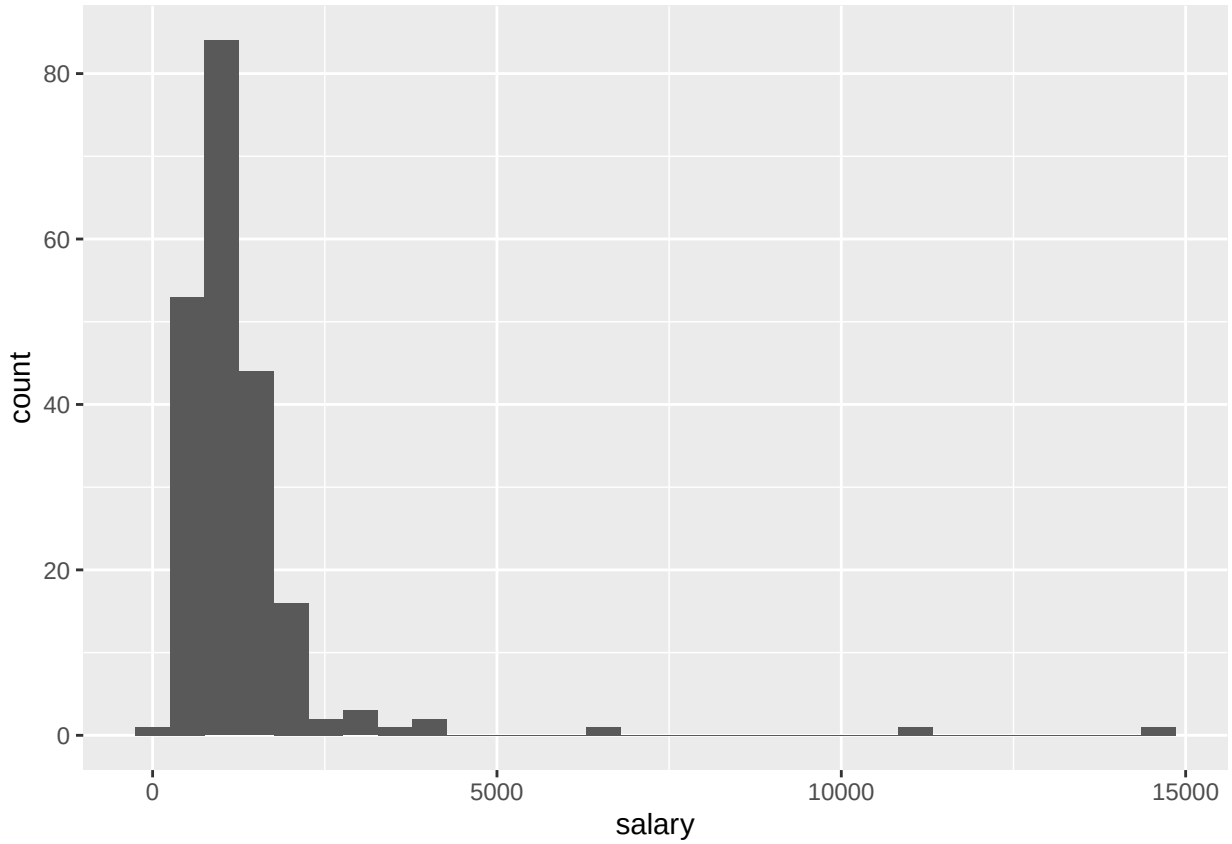


Grammar of Graphics (ggplot2)

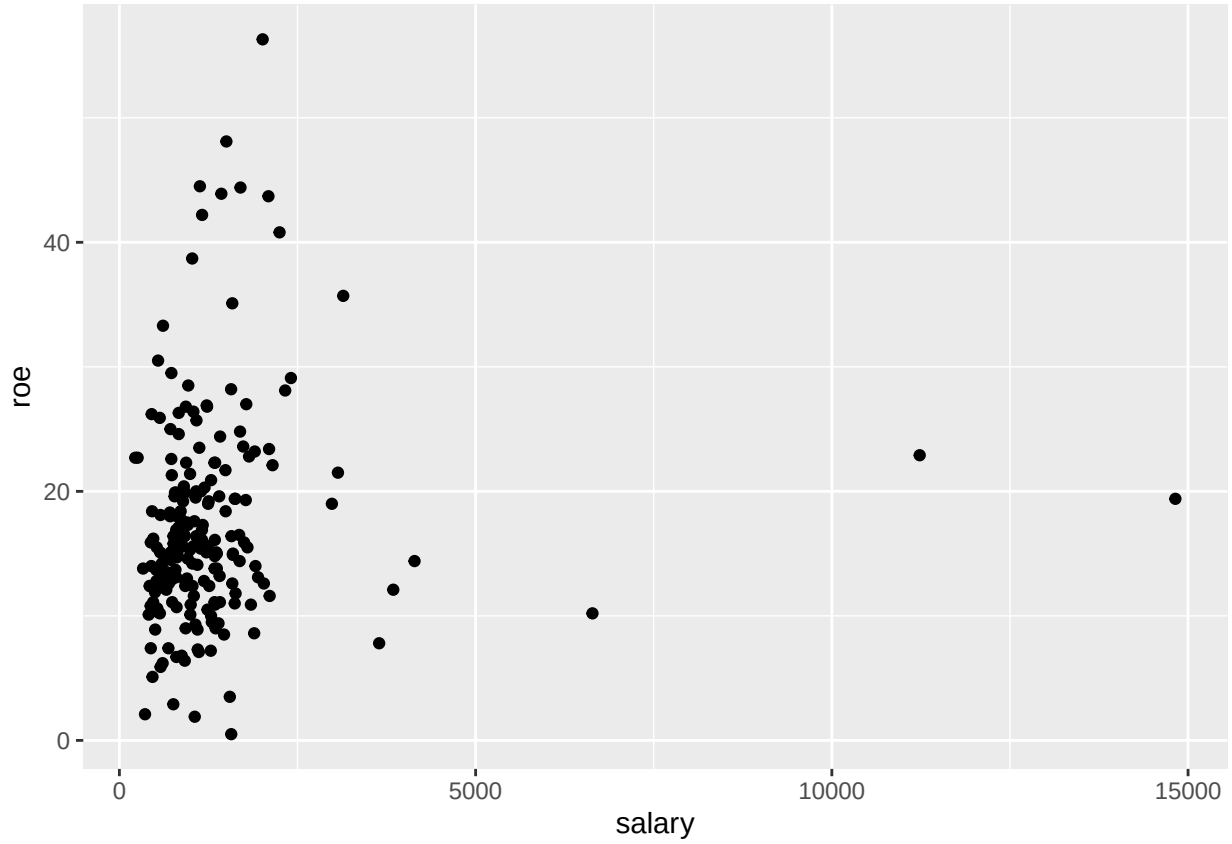
Some people prefer base graphics. Some prefer gg. Depending on what packages you use, you should know a little of each. You can call the ggplot() function with no arguments (but add them in layers) or inline. Either way, the main arguments you need to specify include (1) data and (2) aesthetics (aes).

Try replicating the previous plots using ggplot geom_hist() and geom_point syntax.

```
ggplot() +
  geom_histogram(data = ceosal1, mapping = aes(salary))
```



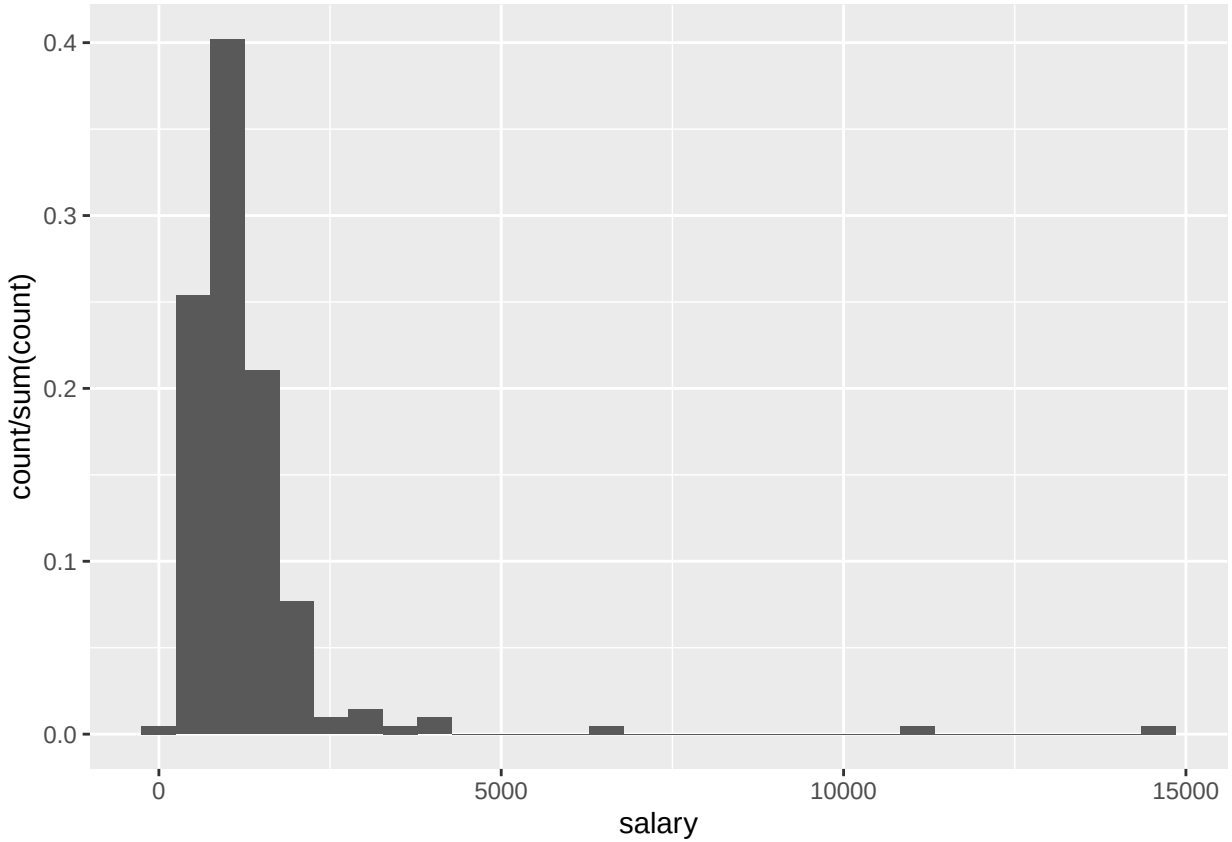

```
ggplot() +
  geom_point(data = ceosal1, mapping = aes(salary, roe))
```



Options in `ggplot`

Implement the suggestion in the feedback.

```
ggplot() +
  geom_histogram(data = ceosal1, mapping = aes(x = salary, y = stat(count)/sum(count)))
```


0.4
0.3
0.2
0.1
0.0
count/sum(count)
0
5000
10000
15000
salary