lab3

Zeren Li

9/18/2019

Roadmap

- OLS
- `ggplot2`

OLS

$$Y_i = \beta_0 + \beta_1 X_i + u_i$$

- the index i runs over the observations, $i = 1, \dots, n$
- Y_i is the *dependent variable*, the *regressand*, or simply the *left-hand variable*
- X_i is the *independent variable*, the *regressor*, or simply the *right-hand variable*
- β_0 is the *intercept* of the population regression line
- β_1 is the *slope* of the population regression line
- u_i is the *error term*.

The OLS Estimator

- The OLS estimator chooses the regression coefficients such that the estimated regression line is as "close" as possible to the observed data points.
- Closeness is measured by the sum of the squared errors made in predicting Y given X. Let b_0 and b_1 be some estimators of β_0 and β_1.
- Then the sum of squared estimation errors can be expressed as

$$\sum_{i=1}^{n} (Y_i - b_0 - b_1 X_i)^2.$$

OLS Estimator, Predicted Values, and Residuals

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2}, \\ \hat{\beta}_0 &= \overline{Y} - \hat{\beta}_1 \overline{X}. \end{aligned}$$

The OLS predicted values $\widehat{Y}_i$ and residuals $\hat{u}_i$ are

$$\widehat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i,$$

$$\hat{u}_i = Y_i - \widehat{Y}_i$$

The estimated intercept $\hat{\beta}_0$, the slope parameter $\hat{\beta}_1$ and the residuals ($\hat{u}_i$) are computed from a sample of n observations of X_i and Y_i, i, ..., n. These are *estimates* of the unknown population intercept (β_0), slope (β_1), and error term (u_i).

Measures of Fit

- How well the model describes the data? The observations are tightly clustered around the regression line
- R^2, the *coefficient of determination*, is the fraction of the sample variance of Y_i that is explained by X_i. Mathematically, the ratio of the explained sum of squares to the total sum of squares.
- The *explained sum of squares* (ESS) is the sum of squared deviations of the predicted values
- $\hat{Y}_i$, from the average of the Y_i The *total sum of squares* (TSS) is the sum of squared deviations of the Y_i from their average. Thus we have
- The sum of squared residuals (SSR) is the sum of squared residuals, a measure for the errors made when predicting the Y by X.

$$\begin{aligned}
ESS &= \sum_{i=1}^{n} \left(\hat{Y}_i - \overline{Y} \right)^2 \\
TSS &= \sum_{i=1}^{n} \left(Y_i - \overline{Y} \right)^2 \\
SSR &= \sum_{i=1}^{n} \left(Y_i - \hat{Y} \right)^2 \\
R^2 &= \frac{ESS}{TSS}
\end{aligned}$$

Returns to Performance

```
ceo %>%
  select(salary, sales) %>%
  summary()
```

```
##      salary           sales
##  Min.   : 100.0   Min.   :   29
##  1st Qu.: 471.0   1st Qu.:  561
##  Median : 707.0   Median : 1400
##  Mean   : 865.9   Mean   : 3529
##  3rd Qu.:1119.0   3rd Qu.: 3500
##  Max.   :5299.0   Max.   :51300
```

```r
# y as dependent variable
y <- ceo$salary

# x as independent variable
x <- ceo$sales

 # beta1
 beta1 = sum((y - mean(y)) * (x - mean(x))) / sum (((x - mean(x))^2))
 beta1
```

```
## [1] 0.03669374
```

```r
 # beta0
 beta0 <- mean(y) - beta1 * mean(x)
 beta0
```

```
## [1] 736.3552
```

```r
 # predicted Y
 y_hat <- beta1 * x + beta0
 head(y_hat)
```

```
## [1]  963.8564  746.7395  742.5565  776.7183  749.2347 1433.5362
```

```r
 # tss
 tss <- sum( (y - mean(y))^2)
 tss
```

```
## [1] 60765965
```

```r
 # ESS
 ess <- sum( (y_hat - mean(y))^2)
 ess
```

```
## [1] 8784947
```

```r
 # SSR
 ssr <- sum( (y- y_hat)^2 )
 ssr
```

```
## [1] 51981017
```

```r
 # R^2
 r_2 = ess/tss
 r_2
```

```
## [1] 0.1445702
```

```r
m1 <- lm(salary ~ sales , ceo)

summary(m1)
```

```
##
## Call:
## lm(formula = salary ~ sales, data = ceo)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -735.4 -340.2 -125.7  236.5 4474.6
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.364e+02  4.738e+01  15.540  < 2e-16 ***
## sales       3.669e-02  6.747e-03   5.438 1.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 545 on 175 degrees of freedom
## Multiple R-squared:  0.1446, Adjusted R-squared:  0.1397
## F-statistic: 29.58 on 1 and 175 DF,  p-value: 1.788e-07
```

Export regression table

```r
stargazer(m1,
          title = "Effect of Sales on Salary",
          covariate.labels = c("Sales"),
          dep.var.labels = c("CEO's Salary"))
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Thu, Sep 19, 2019 - 17:58:13

Table 1: Effect of Sales on Salary

	Dependent variable:
	CEO's Salary
Sales	0.037***
	(0.007)
Constant	736.355***
	(47.384)
Observations	177
R^2	0.145
Adjusted R^2	0.140
Residual Std. Error	545.009 (df = 175)
F Statistic	29.576*** (df = 1; 175)
Note:	*p<0.1; **p<0.05; ***p<0.01

Data Visualization using `ggplot2`

Overview

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Terminology

A statistical graphic is a…

- mapping of **data**
- which may be **statistically transformed** (summarised, log-transformed, etc.)
- to **aesthetic attributes** (color, size, xy-position, etc.)
- using **geometric objects** (points, lines, bars, etc.)
- and mapped onto a specific **facet** and **coordinate system**

Ask yourself these questions before using `ggplot()`

- Which data is used as an input?
- Are the variables statistically transformed before plotting?
- What geometric objects are used to represent the data?
- What variables are mapped onto which aesthetic attributes?
- What type of scales are used to map data to aesthetics?

Anatomy of a ggplot

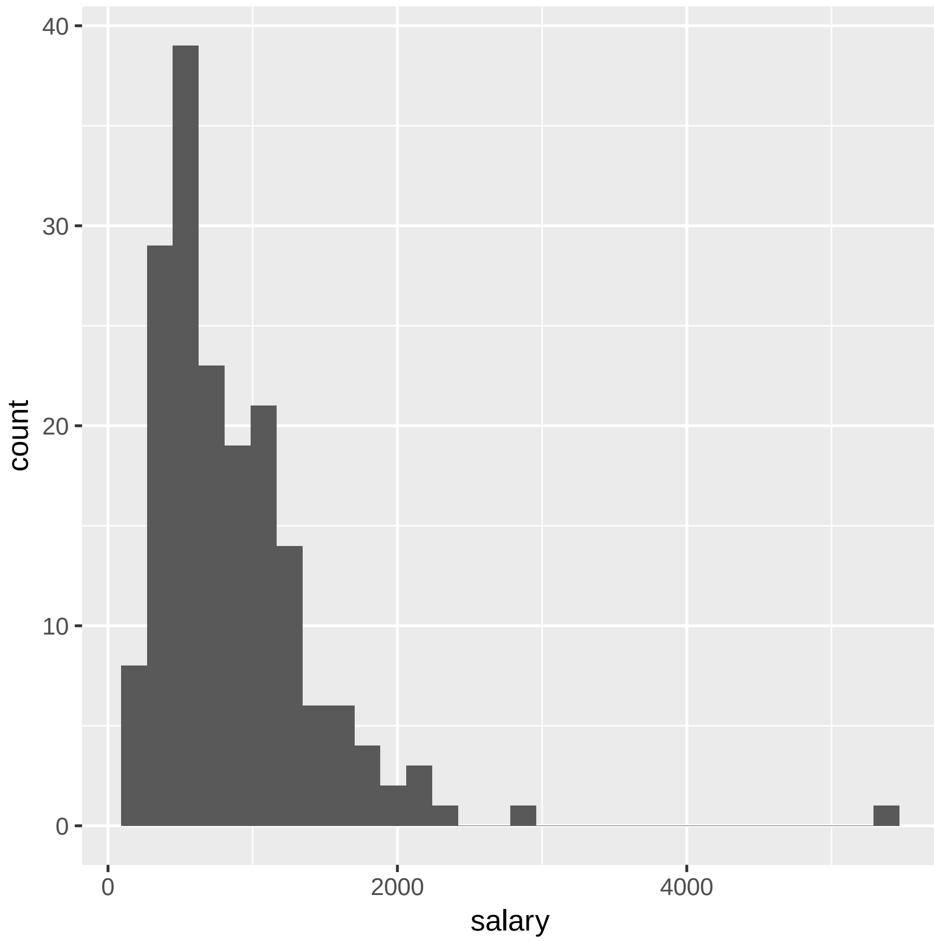
```
ggplot(
  data = [dataframe],
  aes(
    x = [var_x], y = [var_y],
    color = [var_for_color],
    fill = [var_for_fill],
    shape = [var_for_shape]
  )
) +
  geom_[some_geom]([geom_arguments]) +
  ... # other geometries
  scale_[some_axis]_[some_scale]() +
  facet_[some_facet]([formula]) +
  ... # other options
```

Scatterplot - CEO salary and sales

Histogram

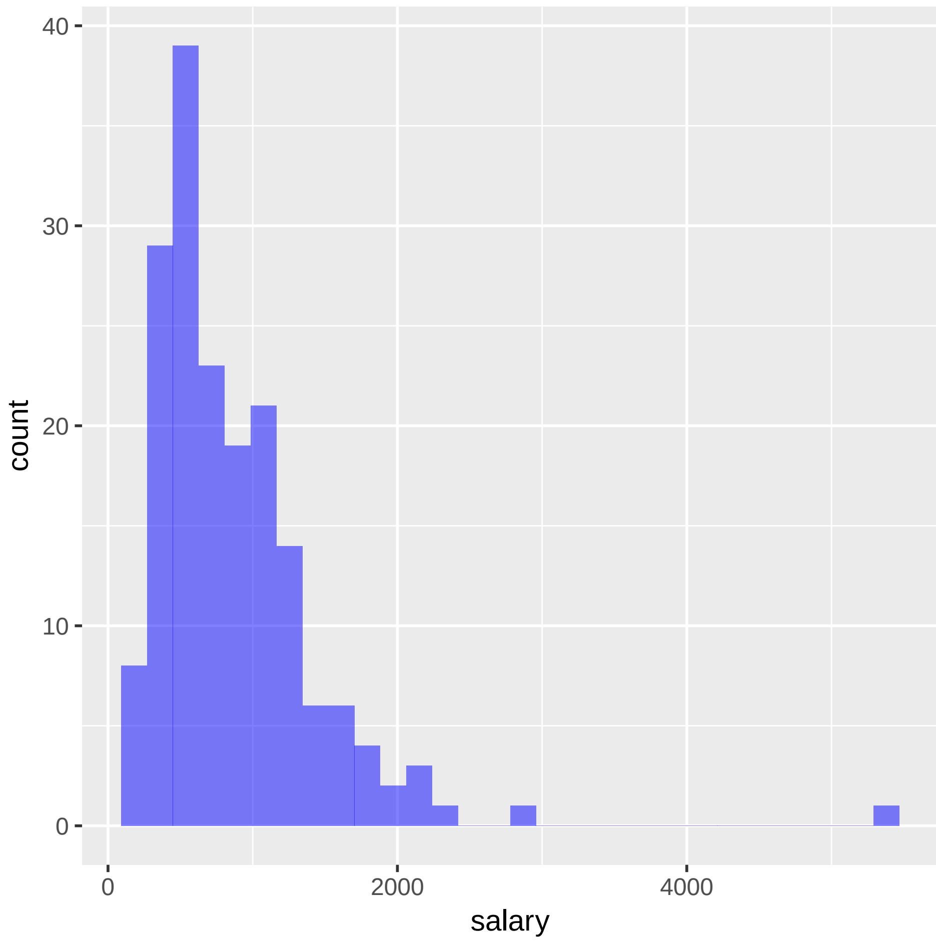
```
ggplot(data = ceo, aes(x = salary))  +
    geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



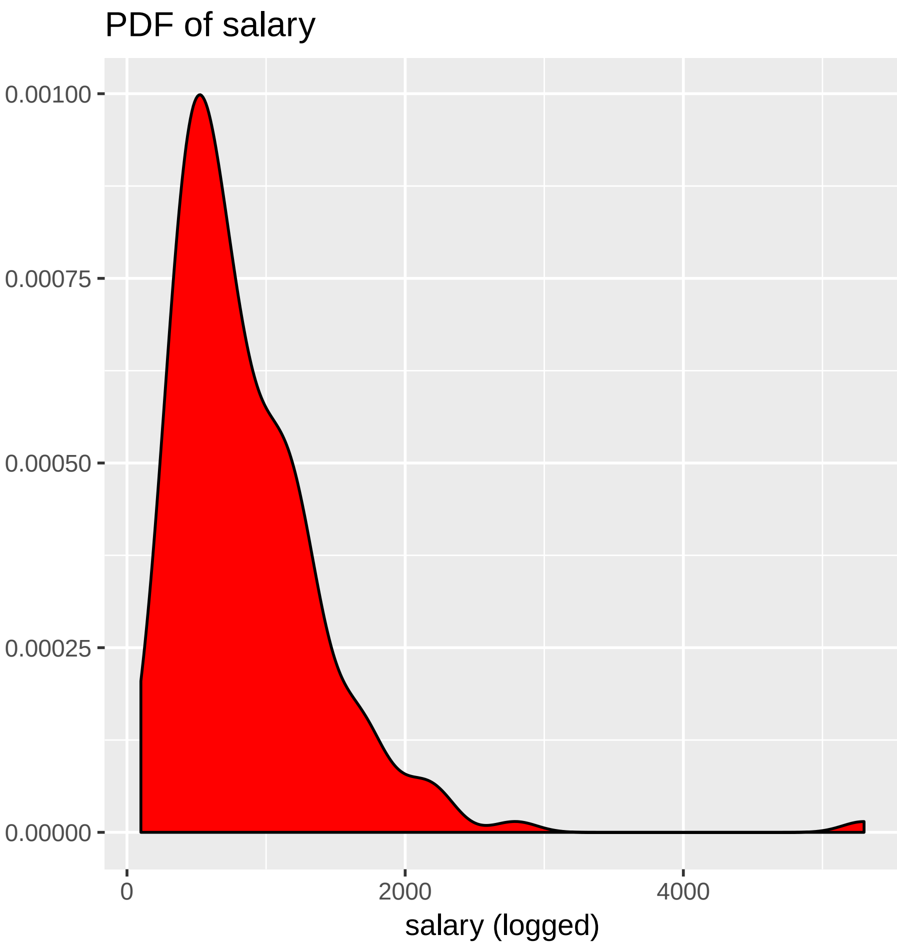

```
ggplot(data = ceo, aes(x = salary)) +
  geom_histogram(alpha = .5, fill = "blue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Distribution

```
ggplot(data = ceo, aes(x = salary))  +
    geom_density(fill = "red")  +
    xlab("salary (logged)") +
    ylab("") +
    ggtitle("PDF of salary")
```

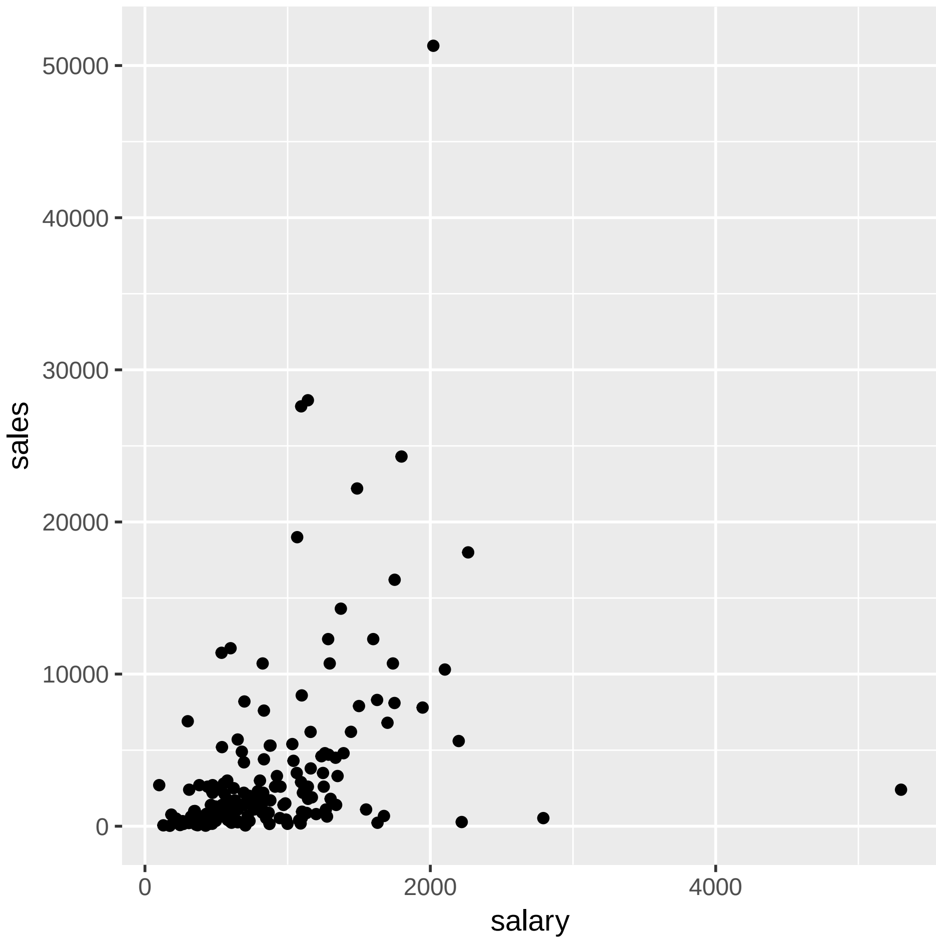



```
ggsave("./hist.pdf")
```

```
## Saving 5 x 5 in image
```

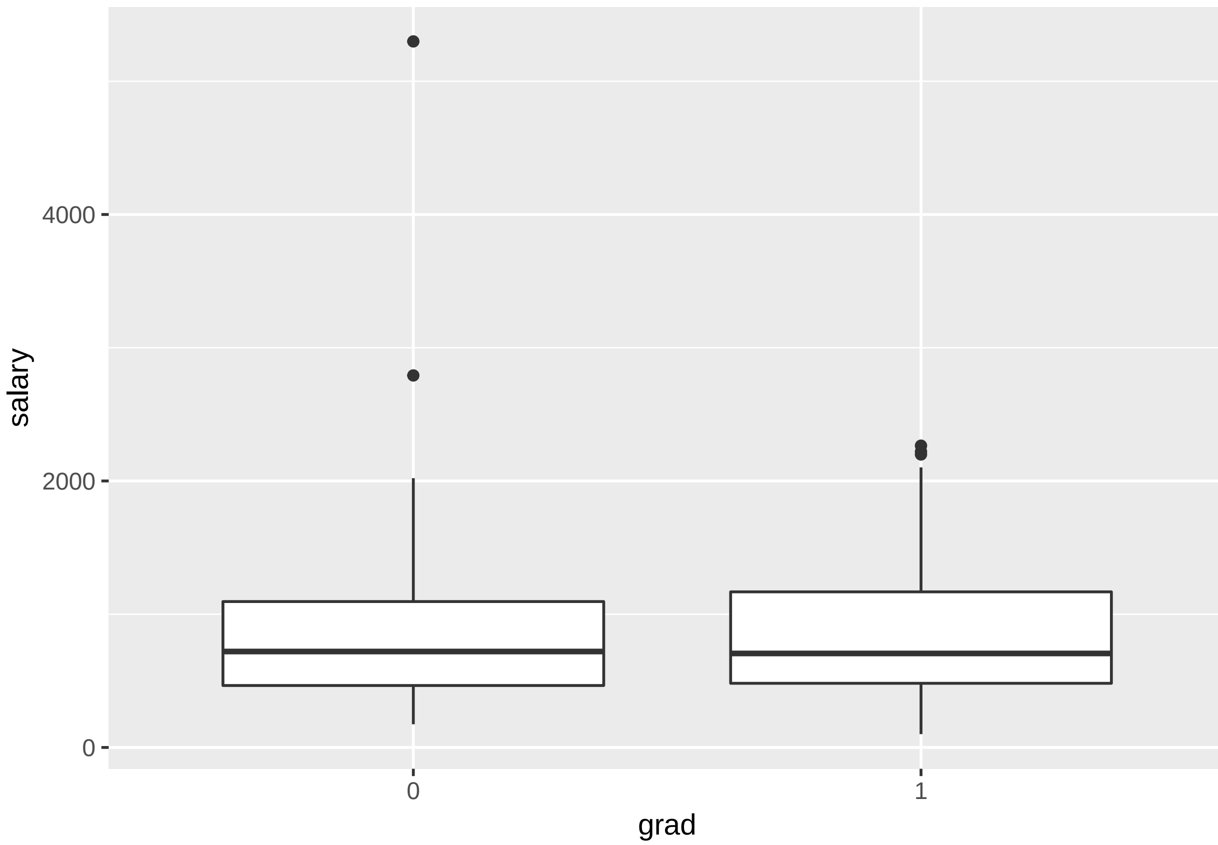
Scatterplot

```
ggplot(data = ceo, aes(x = salary, y = sales)) +
  geom_point()
```



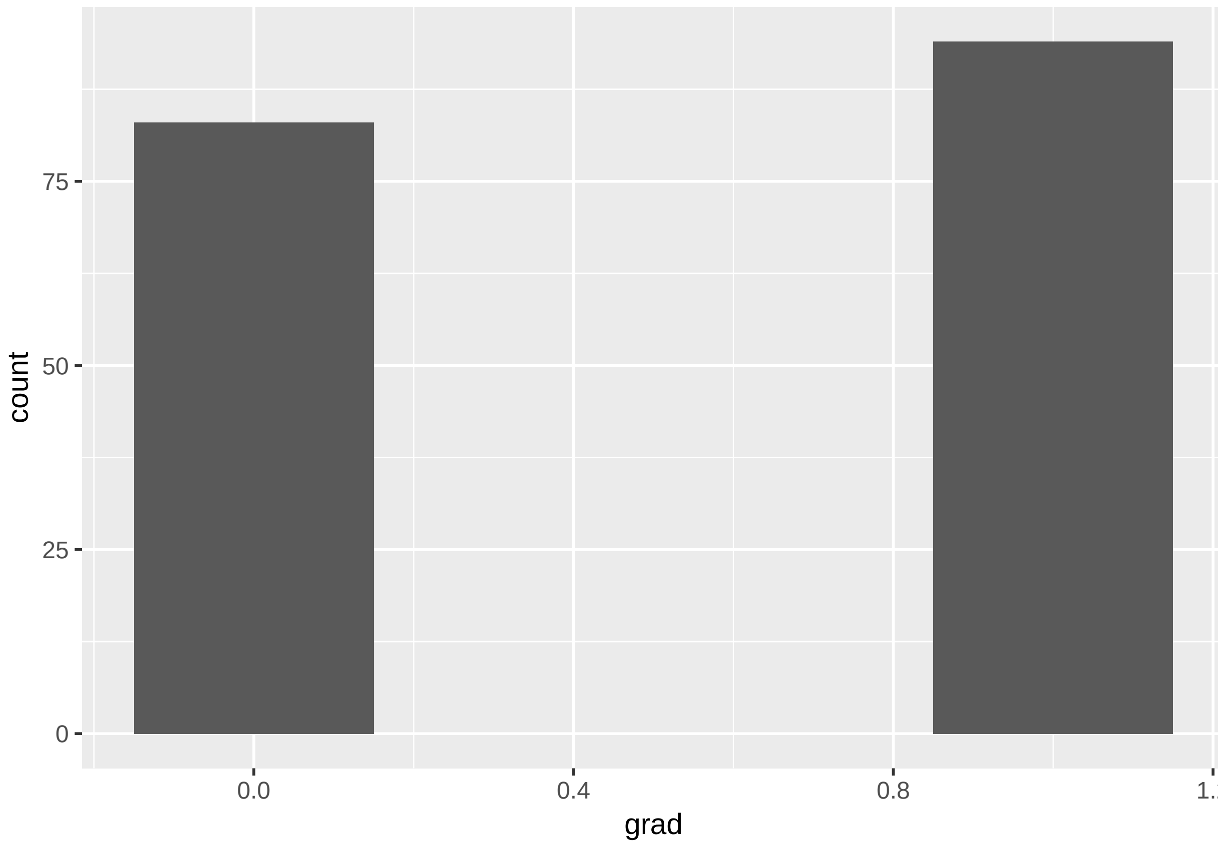
Box plot

```
ceo %>%
  mutate(grad = as.factor(grad)) %>%
ggplot(., aes(x = grad, y = salary )) +
    geom_boxplot()
```



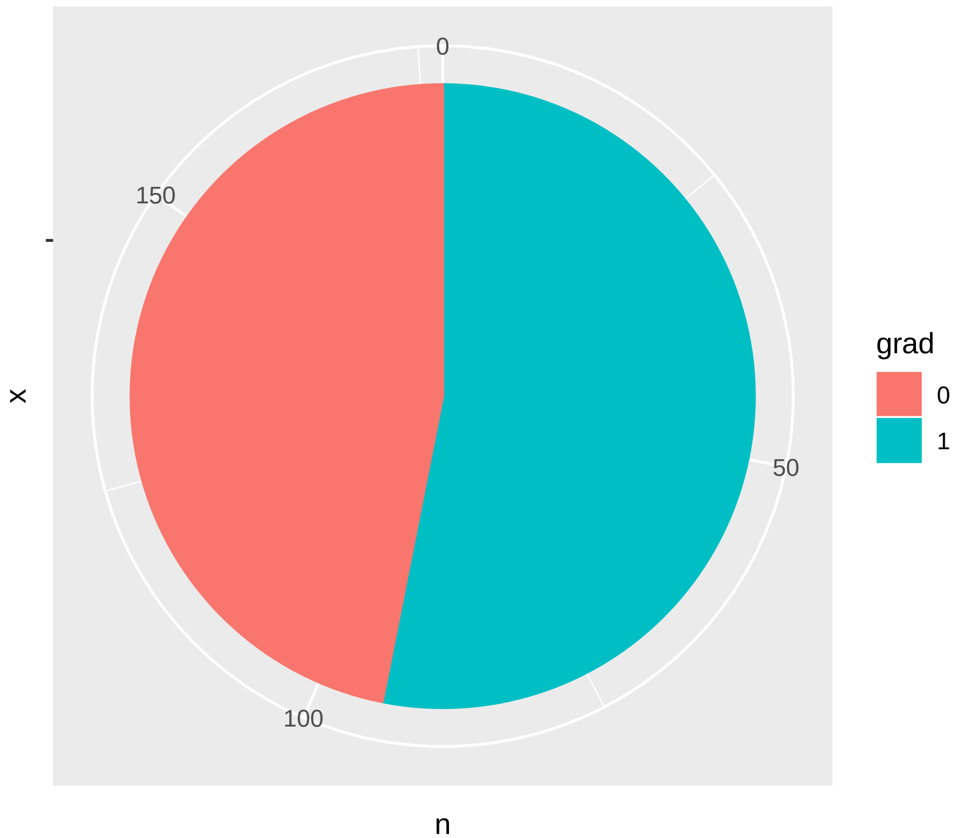
Bar plot

```
ggplot(data = ceo, aes(x = grad )) +
  geom_bar(width = .3)
```



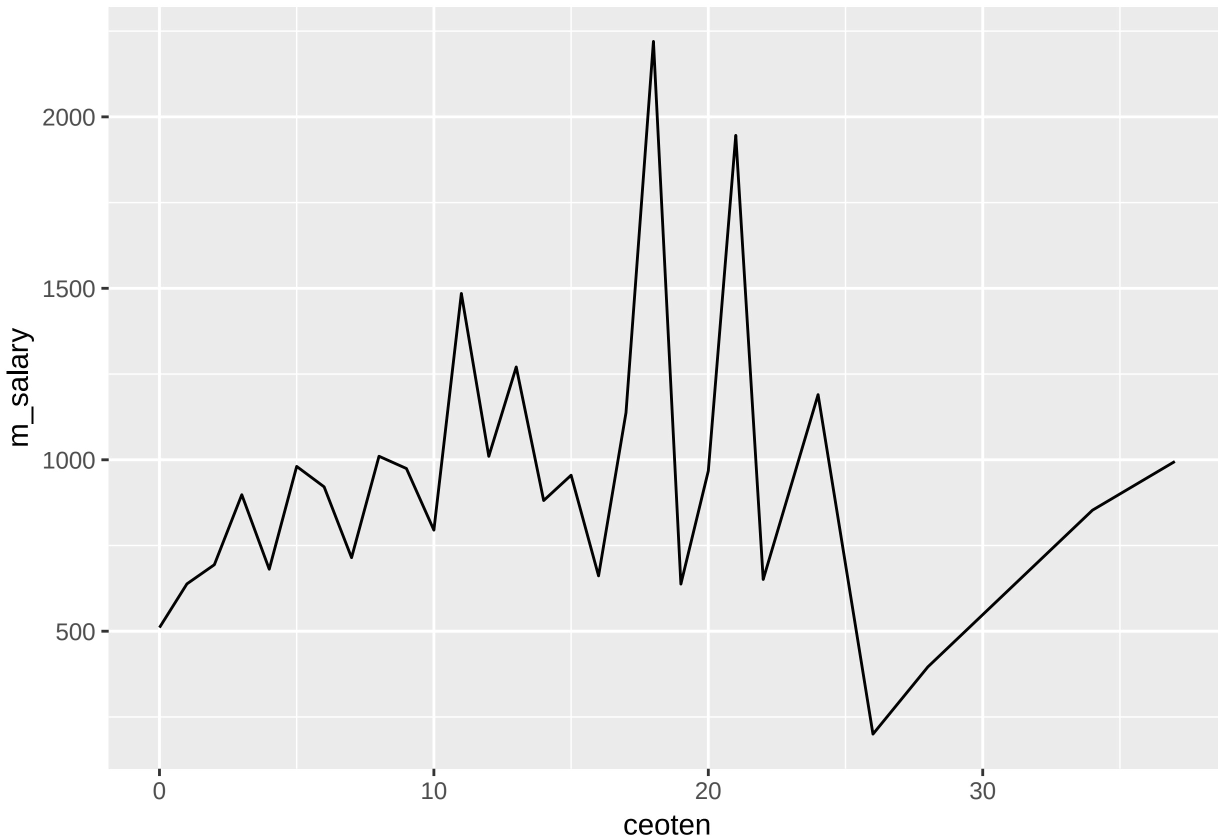
Pie chart

```
# compute mean of salary first
ceo_grad_sum <- ceo %>%
                mutate(grad = as.factor(grad)) %>%
                    group_by(grad) %>%
                    summarize(n = n())

ggplot(data = ceo_grad_sum, aes(x = "", y = n, fill = grad, color = grad) )+
geom_bar(width = 1, stat = "identity")+
  coord_polar("y", start=0)
```



Line chart

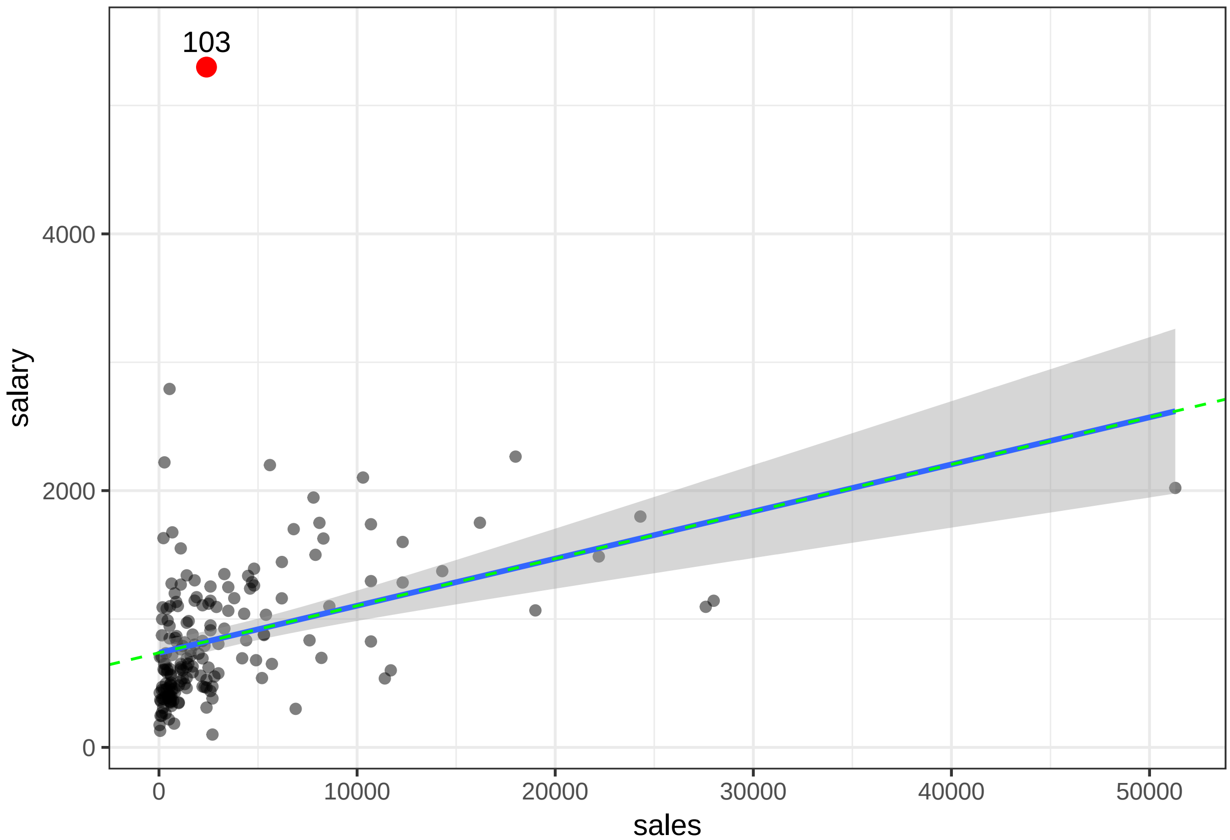
```
# compute mean of salary first
ceo_sum <- ceo %>%
                group_by(ceoten) %>%
                summarise(m_salary = mean(salary, na.rm = T))

ggplot(data = ceo_sum, aes(x = ceoten, y = m_salary )) +
  geom_line()
```



More…

Visualization of OLS

```
outlier <- ceo %>%
  mutate(index = row_number() %>% as.factor() ) %>%
  select(salary, sales,index) %>%
  filter(salary > 4000 )

ggplot(ceo, aes(x = sales, y = salary)) +
  # add points
  geom_point(alpha = .5) +
  # highlight outlier(s)
  geom_point(data = outlier, aes(x=sales,y=salary),
             color = 'red',size=3)  +
  geom_text(data = outlier, aes(x=sales,y= (salary +200)
                                    , label = index))  +
  # add fitted line
  geom_smooth(method = "lm") +
  # add fitted line by hand
  geom_abline(slope = beta1, intercept = beta0, linetype = "dashed", color = "green") +
  # set theme
  theme_bw()
```



Resource

ggplot2 website: https://ggplot2.tidyverse.org/

cheatsheet: https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf

top 50 visualization: http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html