Problem 1 (Sasha Chepurnoi)

Code

Installing packages

Hide

```
#install.packages("readxl")
#install.packages("tidyverse")
#install.packages("ggplot2")
#install.packages("GGally")
library(ggplot2)
library(readxl)
library(tidyverse)
library(dplyr)
library(tidyr)
library(reshape2)
library(GGally)
```

Reading data

Hide

```
ceo <- read_xls("../data/ceo.xls")[,1:7]
head(ceo)
```

salary <dbl>	totcomp <dbl>	tenure <dbl>	age <dbl>	sales <dbl>	profits <dbl>	assets <dbl>
3030	8138	7	61	161315	2956	257389
6050	14530	0	51	144416	22071	237545
3571	7433	11	63	139208	4430	49271
3300	13464	6	60	100697	6370	92630
10000	68285	18	63	100469	9296	355935
9375	42381	6	57	81667	6328	86100

6 rows

Task 1.

a

Salary dataset summary:

Mean is the average salary in the dataset

Min is the lowest salary in the dataset

1st Qu. is the value of the first quantile (0.25). Interpretation: smallest 25% of the salaries of the dataset are less or equal 1084

Median is effectively 0.5 quantile, median value of the salaries (50% of the salaries under value of median)

3rd Qu. is quantile 0.75

Max is the biggest salary value in the dataset

Trimmed 10% mean is 1710. This value is significantly lower than non trimmed mean and could indicate some high outliers in the data, that shift the non trimmed mean to the higher value

Lower 10% quantile shows that 10% of the smallest salaries are less or equal 750, while 10% of the biggest salariest are more than 3384.4

Hide

```
#trimmed mean
sprintf("Trimmed 10%% mean %.2f", mean(ceo$salary, trim=0.1))
```

```
[1] "Trimmed 10% mean 1710.09"
```

Hide

```
#Lowest 10% quantile
sprintf("10%% lower quantile: %.2f", quantile(ceo$salary, prob=0.1))
```

```
[1] "10% lower quantile: 750.00"
```

Hide

```
# Highest 10% quantile
sprintf("10%% upper quantile: %.2f", quantile(ceo$salary, prob=0.9))
```

```
[1] "10% upper quantile: 3384.40"
```

Hide

```
#Summary
summary(ceo$salary)
```

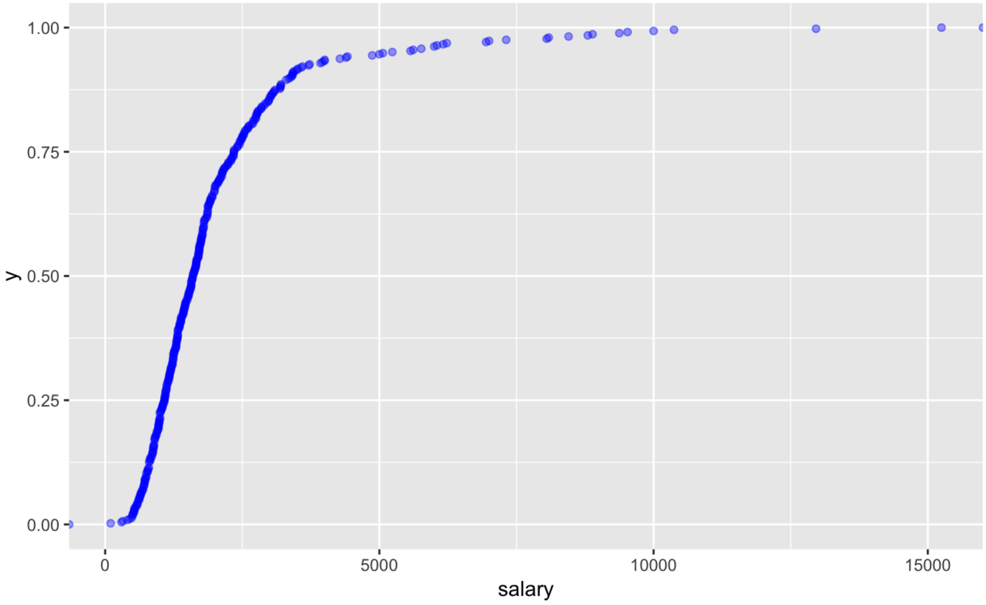
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    100    1084    1600    2028    2348   15250 
```

b

ECDF plot

Hide

```
ggplot(ceo, aes(salary)) + stat_ecdf(geom='point', color='blue', alpha=0.4)
```



$y = \hat{F}^{-1}(0.2)$ Is the 0.2 quantile

$y = \hat{F}^{-1}(0.8)$ Is the 0.8 quantile

$y = \hat{F}(1000)$ Is ecdf of 1000 - result is probability of the values from the distribution to be less of equal 1000. Computed value is 0.224 which indicates that 22.4% is probability of salary to be less than 1000.

$y = 1 - \hat{F}(5000)$ Is probability of the salary value to be higher than 5000. Computed value is 0.053 which indicates that only 5.4% of salaries are higher than 5000!

Hide

```
#b
F = ecdf(ceo$salary)
sprintf("0.2 quantile = %.3f", quantile(ceo$salary, prob=0.2))
```

```
[1] "0.2 quantile = 976.200"
```

Hide

```
sprintf("0.8 quantile = %.3f", quantile(ceo$salary, prob=0.8))
```

```
[1] "0.8 quantile = 2613.000"
```

Hide

```
sprintf("F(1000) = %.3f", F(1000))
```

```
[1] "F(1000) = 0.224"
```

Hide

```
sprintf("1 - F(5000) = %.3f", 1 - F(5000))
```

```
[1] "1 - F(5000) = 0.054"
```

c, d

From the boxplot we can see that location measures that we computed before are still appropriate, because ggplot boxplot is using the same approach.

The documentation says: “The lower and upper hinges correspond to the first and third quartiles (the 25th and 75th percentiles). This differs slightly from the method used by the boxplot function, and may be apparent with small samples. See boxplot.stats() for more information on how hinge positions are calculated for boxplot”.

Default boxplot implementation depends on the number of observations.

Given salary distribution is right-skewed (empirical skewness is >0)

Hide

```
skew <- sum(((ceo$salary - mean(ceo$salary)) / sd(ceo$salary)) ^ 3) / dim(ceo)[1]
sprintf("Skewness: %s", skew)
```

```
[1] "Skewness: 3.37963196696793"
```

Hide

```
sprintf("STD: %.3f", sd(ceo$salary))
```

```
[1] "STD: 1722.566"
```

Hide

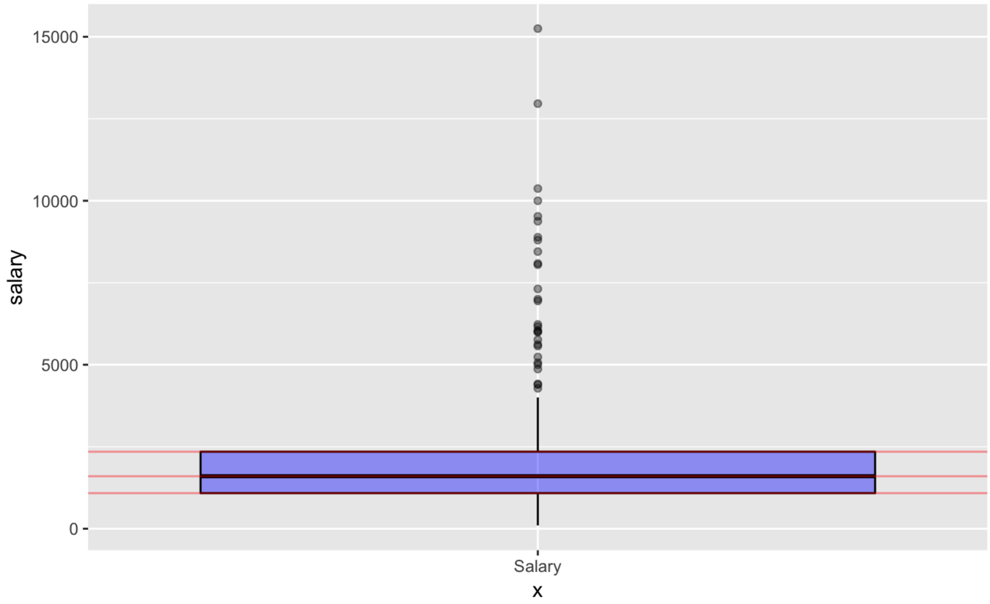
```
sprintf("Variance: %.3f", var(ceo$salary))
```

```
[1] "Variance: 2967234.963"
```

Distribution is right-skewed because mean if shifted to the right of the median. (We can see a lot of outliers with high salary values)

Hide

```
#c
ggplot(ceo) +
    geom_boxplot(aes(x="Salary", y=salary), color='black', fill='blue', alpha=0.4, show.legend = TRUE) +
    geom_hline(yintercept = quantile(ceo$salary, probs = 0.25), color='red', alpha=0.4) +
    geom_hline(yintercept = median(ceo$salary), color='red', alpha=0.4) +
    geom_hline(yintercept = quantile(ceo$salary, probs = 0.75), color='red', alpha=0.4)
```



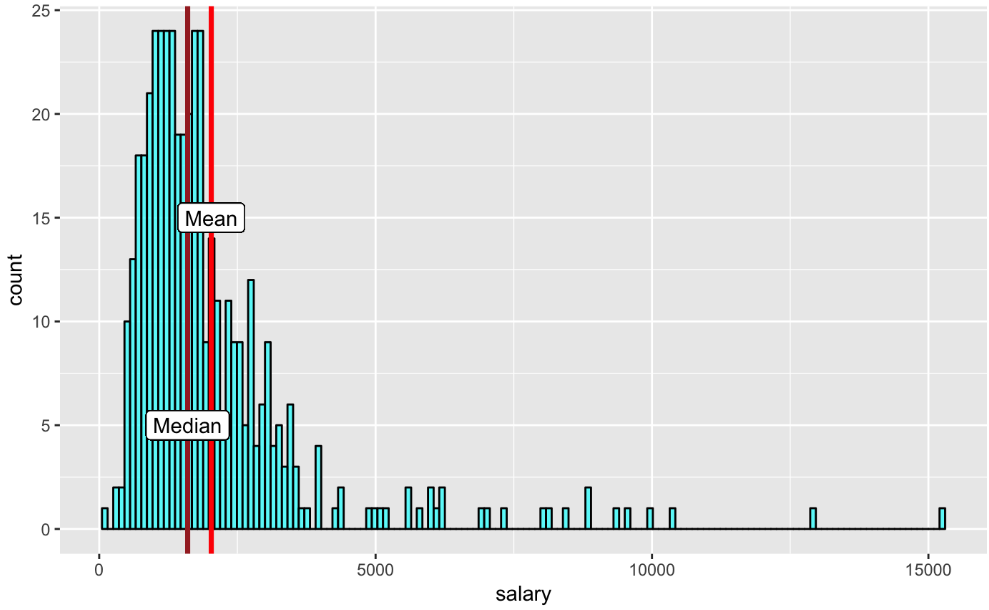
For bin number ggplot uses a fixed number of bins=30. We can regulate bin number or binwidth.

To get the best value for bin/binwidth we should try to explore the data with different values and observe results.

Histogram can help us to approximately estimate the mean and distribution, visualise outliers and data range. With small number of bins we could see the general picture of disribution and trying different bin numbers we could find interesting patterns in data.

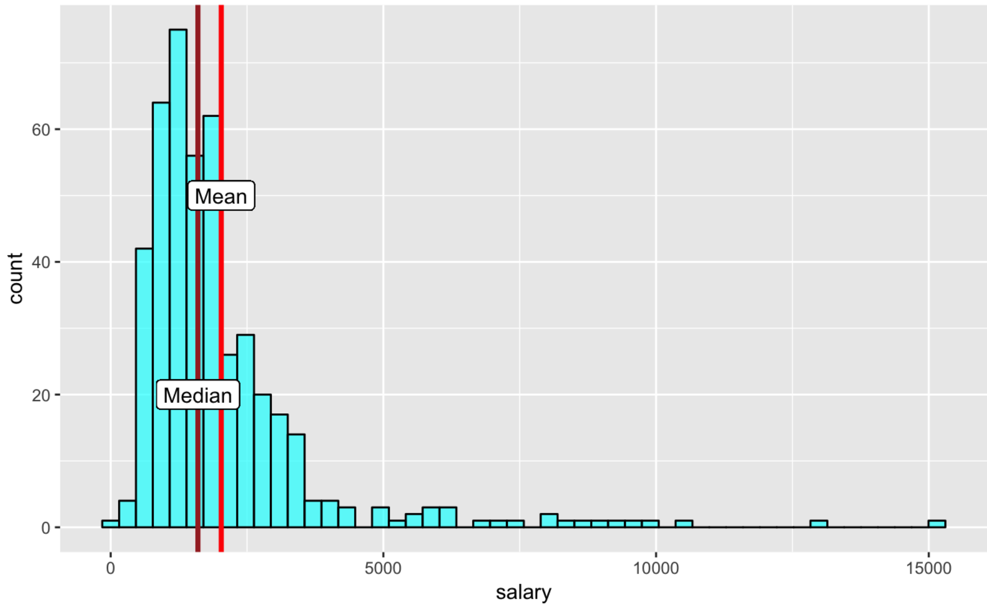
Hide

```
s_median = median(ceo$salary)
s_mean = mean(ceo$salary)
mean_line = geom_vline(xintercept = s_mean, color='red', size=1.25)
median_line = geom_vline(xintercept = s_median, color='brown', size=1.25)
ggplot(ceo) + geom_histogram(aes(x=salary), bins = 150, fill='cyan', color='black', alpha=0.7) +
  mean_line +
  median_line +
  geom_label(x=s_mean, y=15, label='Mean') +
  geom_label(x=s_median, y=5, label='Median')
```



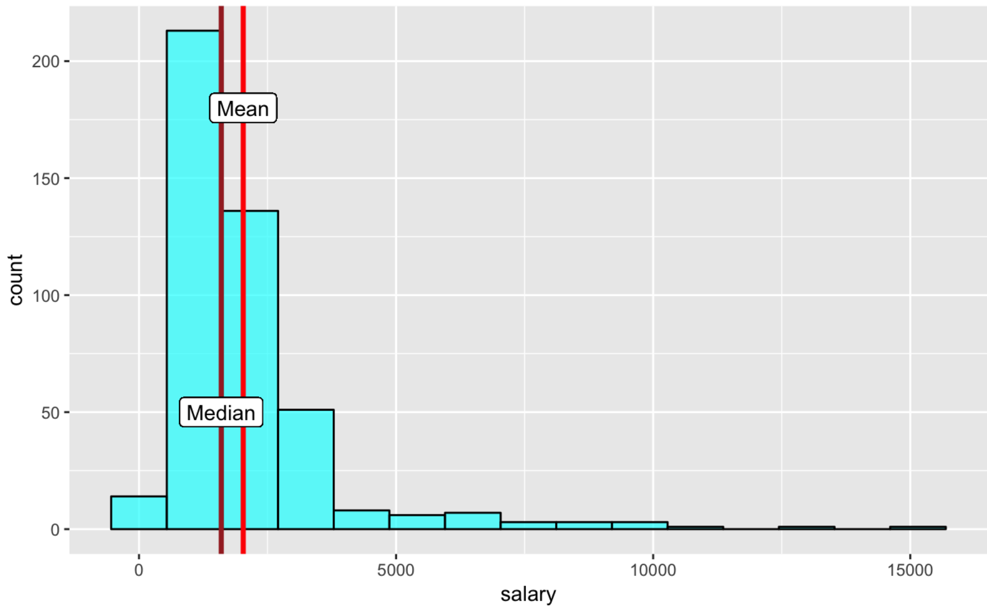
Hide

```
ggplot(ceo) + geom_histogram(aes(x=salary), bins = 50, fill='cyan', color='black', alpha=0.7)  +
  mean_line +
  median_line +
  geom_label(x=s_mean, y=50, label='Mean') +
  geom_label(x=s_median, y=20, label='Median')
```



Hide

```
ggplot(ceo) + geom_histogram(aes(x=salary), bins = 15, fill='cyan', color='black', alpha=0.7)  +
  mean_line +
  median_line +
  geom_label(x=s_mean, y=180, label='Mean') +
  geom_label(x=s_median, y=50, label='Median')
```



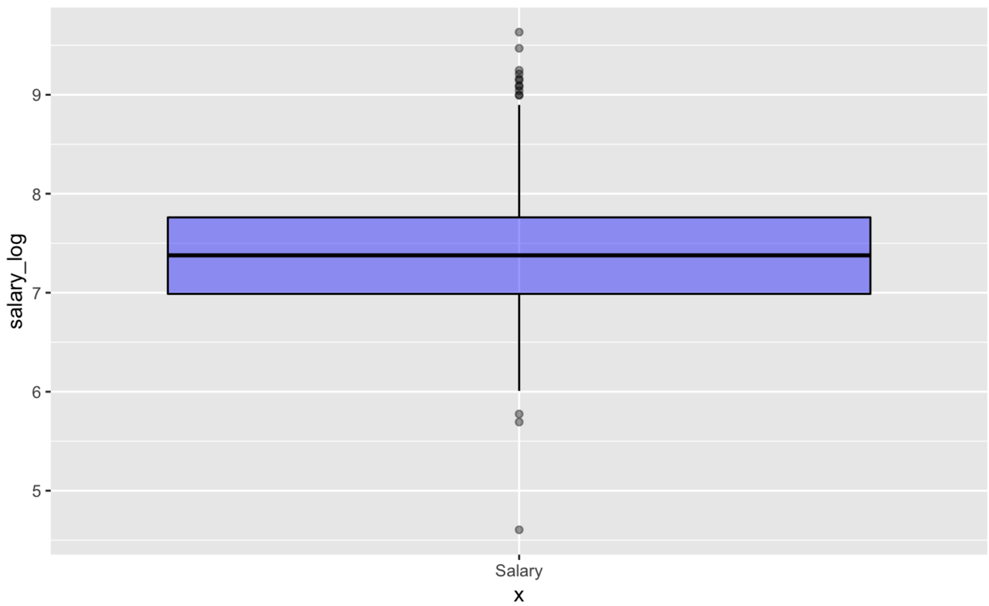
e

After applying log to the data, mean and median are much closer and distribution is less skewed. Variance of the data is also reduced significantly

Applying log to the data can help to deal with huge outliers, but we still have 0 outlier because log function can’t help with 0 values.

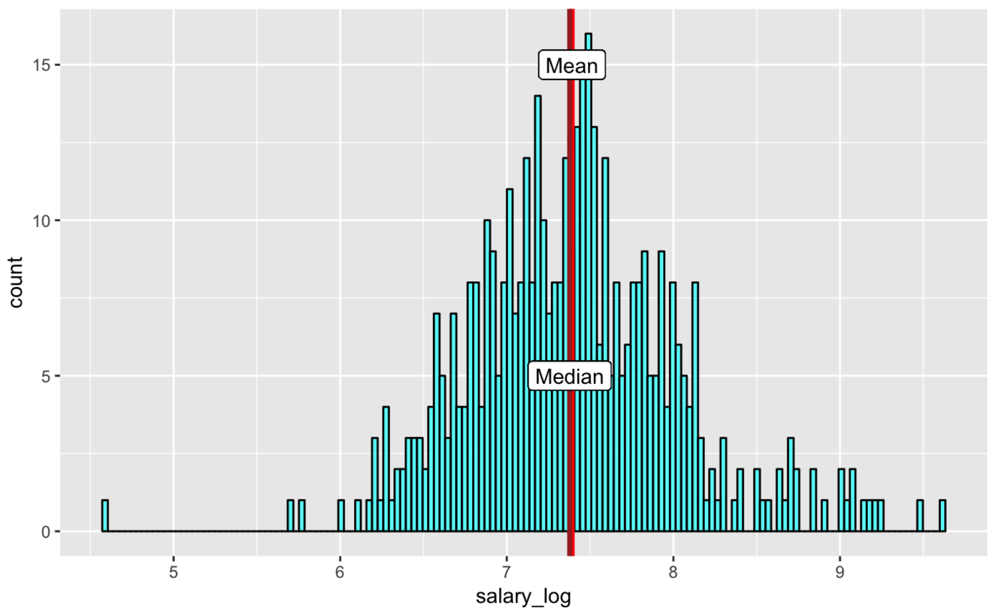
Hide

```
#e
ceo_log = ceo %>% mutate(salary_log=log(salary))
l_mean = mean(ceo_log$salary_log)
l_median = median(ceo_log$salary_log)
mean_line = geom_vline(xintercept = l_mean, color='red', size=1.25)
median_line = geom_vline(xintercept = l_median, color='brown', size=1.25)
ggplot(ceo_log) +
  geom_boxplot(aes(x="Salary", y=salary_log), color='black', fill='blue', alpha=0.4)
```



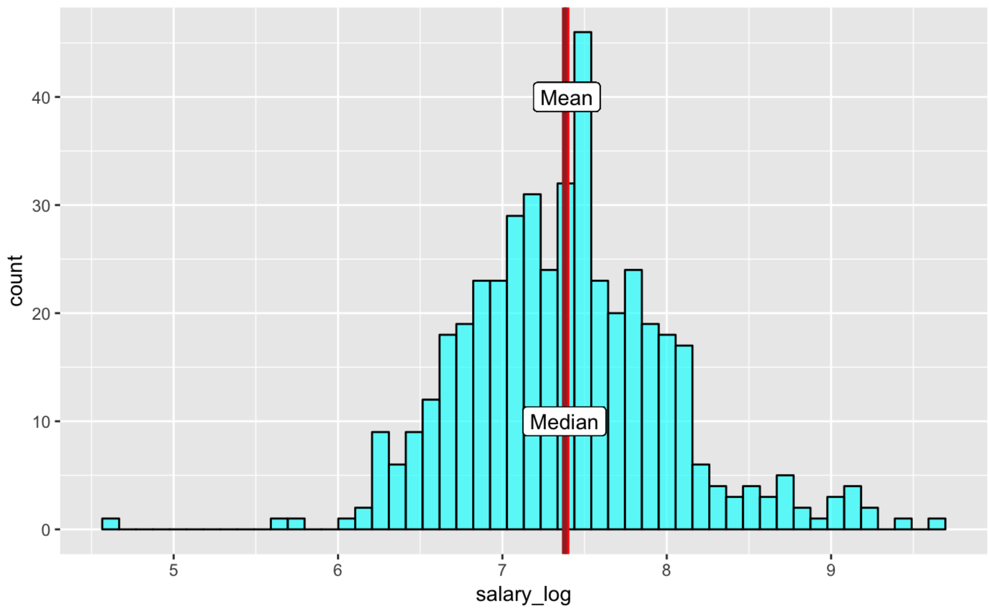
Hide

```
ggplot(ceo_log) + geom_histogram(aes(x=salary_log), bins = 150, fill='cyan', color='black', alpha=0.7) +
  mean_line +
  median_line +
  geom_label(x=l_mean, y=15, label='Mean') +
  geom_label(x=l_median, y=5, label='Median')
```



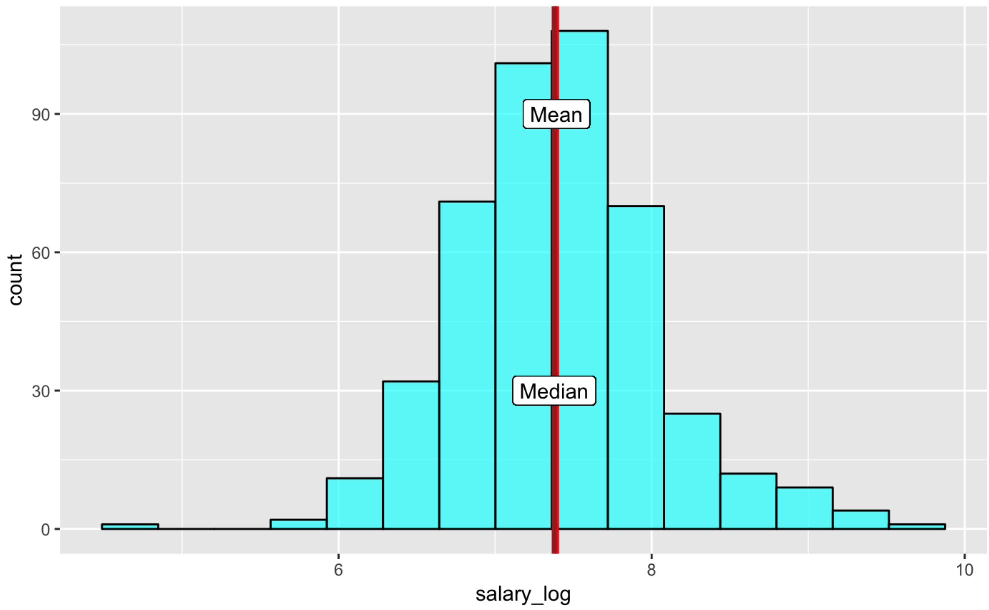
Hide

```
ggplot(ceo_log) + geom_histogram(aes(x=salary_log), bins = 50, fill='cyan', color='black', alpha=0.7) +
  mean_line +
  median_line +
  geom_label(x=l_mean, y=40, label='Mean') +
  geom_label(x=l_median, y=10, label='Median')
```



Hide

```
ggplot(ceo_log) + geom_histogram(aes(x=salary_log), bins = 15, fill='cyan', color='black', alpha=0.7) +
  mean_line +
  median_line +
  geom_label(x=l_mean, y=90, label='Mean') +
  geom_label(x=l_median, y=30, label='Median')
```



Hide

```
skew <- sum(((ceo_log$salary_log - mean(ceo_log$salary_log)) / sd(ceo_log$salary_log)) ^ 3) / dim(ceo_log)[1]
sprintf("Skewness: %s", skew)
```

```
[1] "Skewness: 0.303710747356108"
```

Task 2.

a

From the correlation matrix we can see that we have some important correlations.

Highly correlated variables with salary: sales, profits, assets, totcomp.

All these values are highly releated to the economical value of the company and also highly correlate with salary. Also, we can detect squares of correlation on the plot. **salary-totcomp** square, **tenure-age** square and **sales-profits-assets** square.

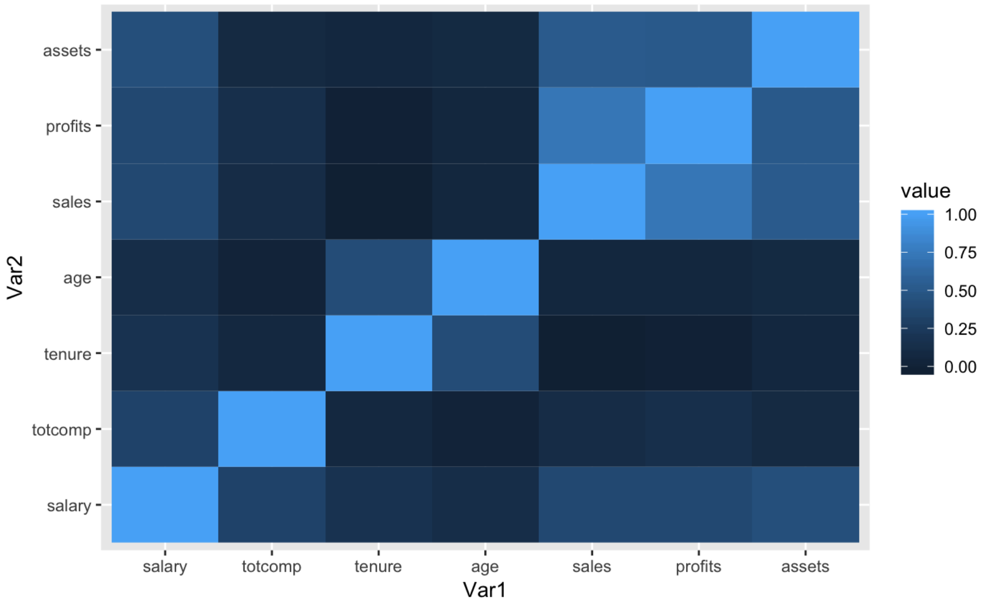
Hide

```
ceo_corr <- ceo %>% cor
ceo_corr
```

```
           salary    totcomp       tenure        age       sales       profits      assets
salary  1.0000000 0.32202331  0.172448581 0.11946557  0.37125562  0.370315210 0.43103599
totcomp 0.3220233 1.00000000  0.067997911 0.01604635  0.10599852  0.139931146 0.08323729
tenure  0.1724486 0.06799791  1.000000000 0.40590101 -0.02774297 -0.006262663 0.05768246
age     0.1194656 0.01604635  0.405901015 1.00000000  0.05035029  0.049095817 0.08118537
sales   0.3712556 0.10599852 -0.027742970 0.05035029  1.00000000  0.725970143 0.51821425
profits 0.3703152 0.13993115 -0.006262663 0.04909582  0.72597014  1.000000000 0.51088896
assets  0.4310360 0.08323729  0.057682460 0.08118537  0.51821425  0.510888960 1.00000000
```

Hide

```
ceo_corr_melt <- ceo_corr %>% melt
ggplot(data = ceo_corr_melt, aes(x=Var1, y=Var2, fill=value)) + geom_tile()
```

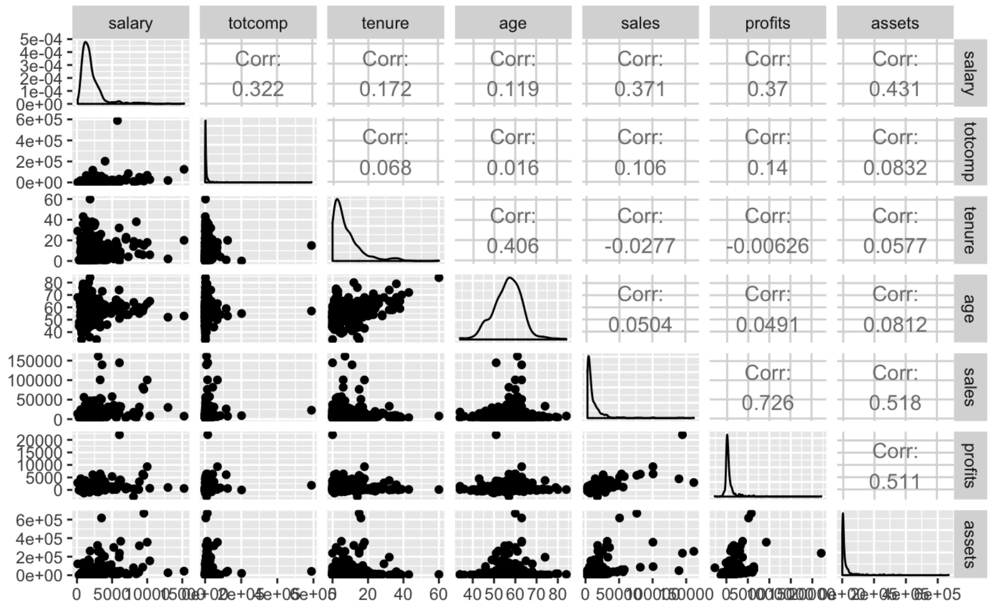


b

From the pairs scatter plot, we can see that pairs with high correlation have noticeable linear dependency in the data.

Hide

```
ggpairs(ceo, 1:7, progress = FALSE)
```



Spearman correlation results shows us that we missed another one square of correlations in the data comparing to Pearson. We can see that **assets-profits-sales** and **salary-totcomp** are also highly correlated.

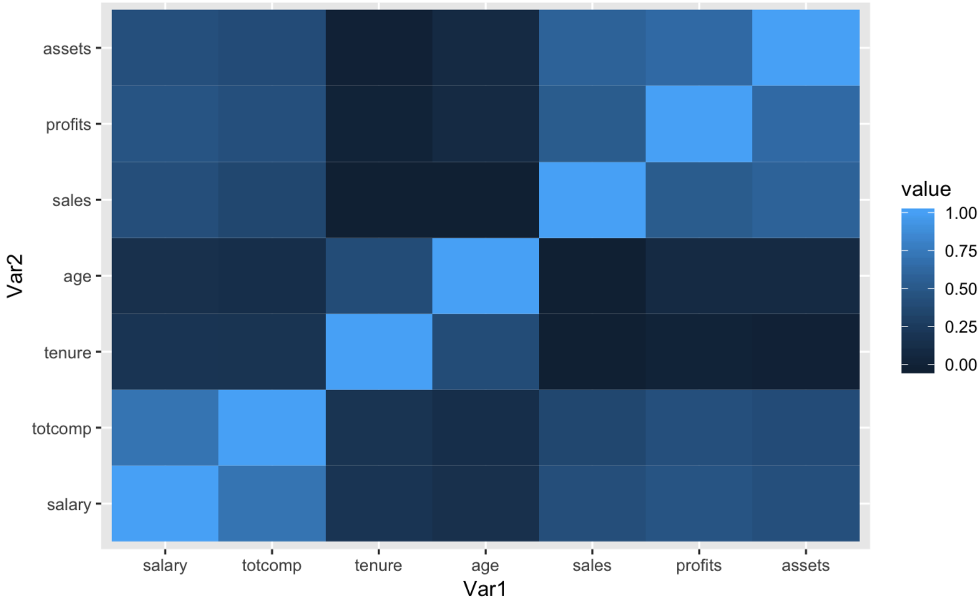
Hide

```
ceo_corr <- ceo %>% cor(method='spearman')
ceo_corr
```

```
           salary   totcomp       tenure         age       sales     profits      assets
salary  1.0000000 0.7078308  0.191974014  0.14926921  0.42109117 0.465469470  0.43100861
totcomp 0.7078308 1.0000000  0.186275787  0.12718939  0.35765620 0.433552007  0.40211661
tenure  0.1919740 0.1862758  1.000000000  0.40298042 -0.02930116 0.005143434 -0.00893509
age     0.1492692 0.1271894  0.402980423  1.00000000 -0.03071638 0.084713887  0.08449035
sales   0.4210912 0.3576562 -0.029301160 -0.03071638  1.00000000 0.527095403  0.58152898
profits 0.4654695 0.4335520  0.005143434  0.08471389  0.52709540 1.000000000  0.63552376
assets  0.4310086 0.4021166 -0.008935090  0.08449035  0.58152898 0.635523760  1.00000000
```

Hide

```
ceo_corr_melt <- ceo_corr %>% melt
ggplot(data = ceo_corr_melt, aes(x=Var1, y=Var2, fill=value)) + geom_tile()
```



Rank is the index of the value in the ordered array.

Using the default rank implementation let's compute the rank of the salary value of 6000: Rank value *429* and *430* as we have multiple entries of salary equals 6000. Rank values for duplicated salaries may vary when using different ties.method.

Documentation:

"If all components are different (and no NAs), the ranks are well defined, with values in seq_along(x). With some values equal (called 'ties'), the argument ties.method determines the result at the corresponding indices. The"first" method results in a permutation with increasing values at each index set of ties, and analogously "last" with decreasing values. The "random" method puts these in random order whereas the default, "average", replaces them by their mean, and "max" and "min" replaces them by their maximum and minimum respectively, the latter being the typical sports ranking."

Hide

```
rank(ceo$salary, ties.method = 'first')[ceo$salary == 6000]
```

```
[1] 429 430
```

c

Splitting dataframe into older/younger CEOs

Hide

```
younger = ceo %>% filter(age < 50)
F_y = ecdf(younger$salary)
older = ceo %>% filter(age >=50)
F_o = ecdf(older$salary)
```

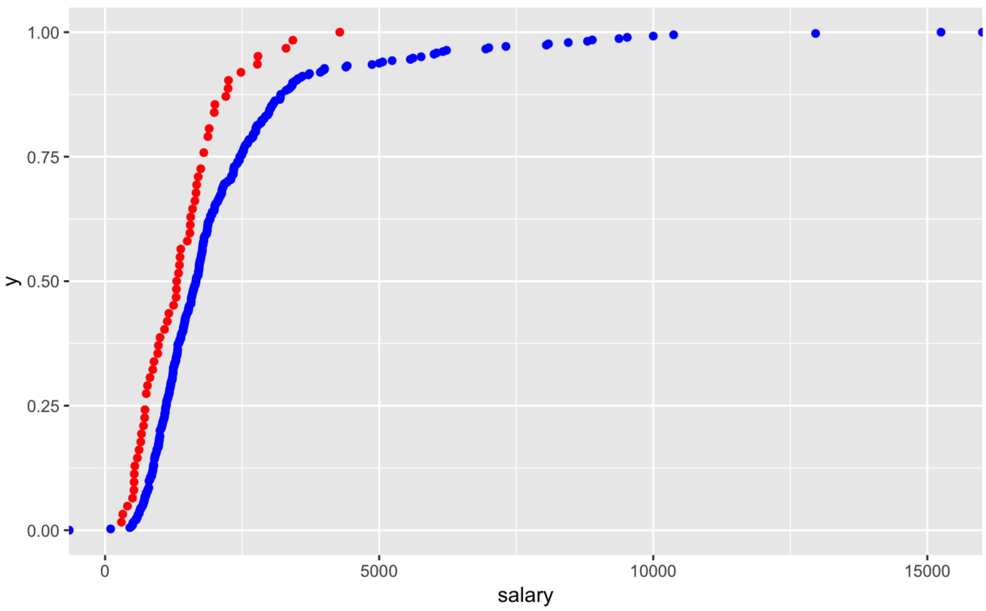
After splitting data into 2 groups we can see significant difference between these groups.

For older CEO group we see much higher std value, higher median, higher mean value and some amount of high salary outliers. This could tell us that older CEOs tend to earn more than younger CEOs.

For younger CEOs we have much smalled std for salaries and smalled mean/median values.

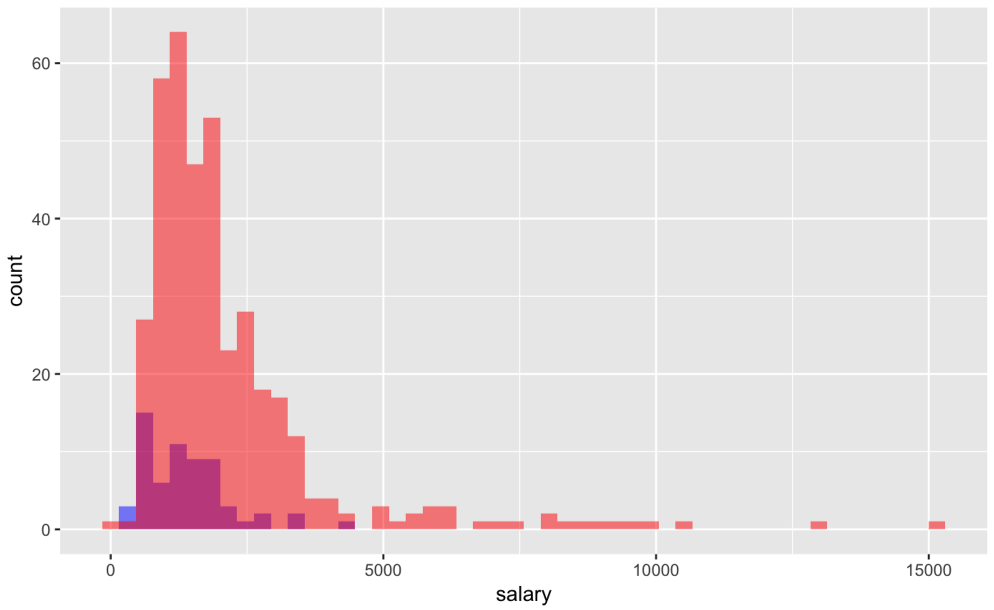
Hide

```
ggplot() +
  stat_ecdf(data=younger, mapping=aes(x=salary), color='red',geom='point') +
  stat_ecdf(data=older, mapping=aes(x=salary), color='blue',geom='point')
```



Hide

```
ggplot() +
  geom_histogram(data=younger, mapping=aes(x=salary), fill='blue', alpha=0.5, bins=50) +
  geom_histogram(data=older, mapping=aes(x=salary), fill='red', alpha=0.5, bins=50)
```



Hide

```
sprintf("Older std: %.3f", sd(older$salary))
```

```
[1] "Older std: 1808.578"
```

Hide

```
sprintf("Younger std: %.3f", sd(younger$salary))
```

```
[1] "Younger std: 805.548"
```

Hide

```
summary(older$salary)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    100    1117    1660    2128    2460   15250 
```

Hide

```
summary(younger$salary)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    297     750    1321    1407    1800    4280 
```

Task 3:

a

Hide

```
salary_groups = c(0, 2000, 4000, max(ceo$salary))
age_groups = c(0, 50, max(ceo$age))
S = ceo %>%
  mutate(salary_group = cut(salary, breaks=salary_groups, include.lowest = TRUE, right = FALSE)) %>%
  mutate(age_group = cut(age, breaks=age_groups, include.lowest = TRUE, right=FALSE))
salary_age_cor <- S %>%
  group_by(salary_group, age_group) %>%
  summarise(c = cor(salary, age)) %>%
  spread(age_group, c)
crosstab <- table(S$age_group, S$salary_group)
S_abs <-  crosstab %>% addmargins
S_rel <- prop.table(crosstab) %>% addmargins
```

Table with absoulte frequencies

Hide

```
S_abs
```

```
         
          [0,2e+03) [2e+03,4e+03) [4e+03,1.52e+04] Sum
  [0,50)         52             9                1  62
  [50,84]       248           107               30 385
  Sum           300           116               31 447
```

Table with relative frequencies

Hide

```
S_rel
```

```
           [0,2e+03) [2e+03,4e+03) [4e+03,1.52e+04]         Sum
  [0,50)  0.116331096   0.020134228      0.002237136 0.138702461
  [50,84] 0.554809843   0.239373602      0.067114094 0.861297539
  Sum     0.671140940   0.259507830      0.069351230 1.000000000
```

b

$n_{12} = 9$ - absoule value of CEOs with age less 50 and salary between 2000 and 4000

$h_{12} = 0.02$ - relative value of the CEO with age less 50 and salary beween 2000 and 4000 (2% from total!)

$n_{1.} = 62$ - absoulte value of CEOs under 50

$h_{1.} = 0.13$ - relative value of the CEOs under 50 (13%)

c

Computing dependency measure between A and S groups (C_{corr})

$C_{corr} = 0.205$ shows us the weak correlation between A and S group.

From this value we can conclude that higher age of the CEO can result in higher salary.

Hide

```
nrow <- dim(crosstab)[1]
ncol <- dim(crosstab)[2]
chi_squared <- 0
for (row in seq(nrow)) {
  for (col in seq(ncol)) {
      n_aprox = S_abs[row, ncol + 1] * S_abs[nrow + 1, col] / S_abs[nrow + 1, ncol + 1]
      chi = ((S_abs[row, col] - n_aprox) ^ 2) / (n_aprox)
      chi_squared <- chi_squared + chi

  }
}
c = sqrt(chi_squared / (chi_squared + S_abs[nrow + 1, ncol + 1]))
c_max = sqrt((min(nrow, ncol) - 1) / min(nrow, ncol))
c_corr = c / c_max
sprintf("C_corr = %.3f", c_corr)
```

```
[1] "C_corr = 0.205"
```