Examining Data with Plots

Youmi Suk

School of Data Science, University of Virginia

1. 24. 2022

Examining Data with Plots

- Plotting Univariate Data
    - Barplots
    - Histogram & kernel density estimates
    - Cumulative distribution fct & quantile-comparison plots
    - Boxplots
- Plotting Bivariate/Multivariate Data
    - Scatterplots
    - Conditioning plots

Examining Data with Plots

Statistical graphs are central to effective data analysis, both in the early stages of an investigation and in statistical modeling.

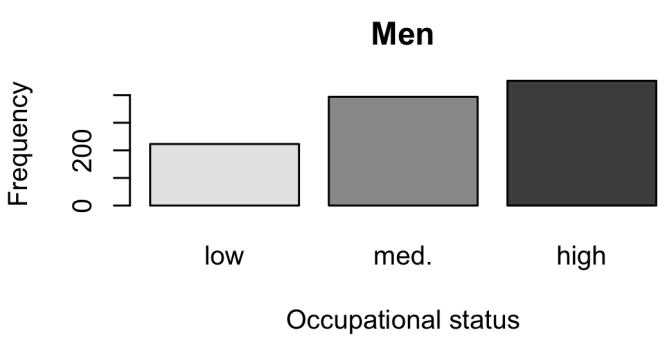
- Fox (2008). Applied Regression Analysis.

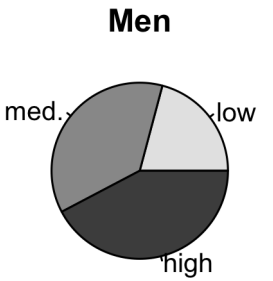
The greatest value of a picture is when it forces us to notice what we never expected to see.

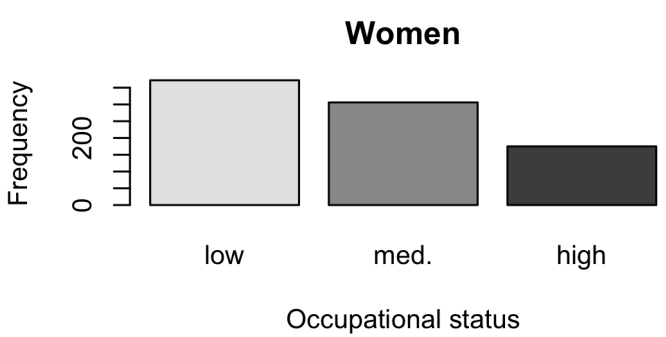
- John Tukey (1977): Exploratory Data Analysis.

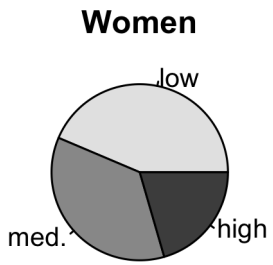
Visualization of a Discrete Variable's Frequency Distribution

Barplot & Pie chart



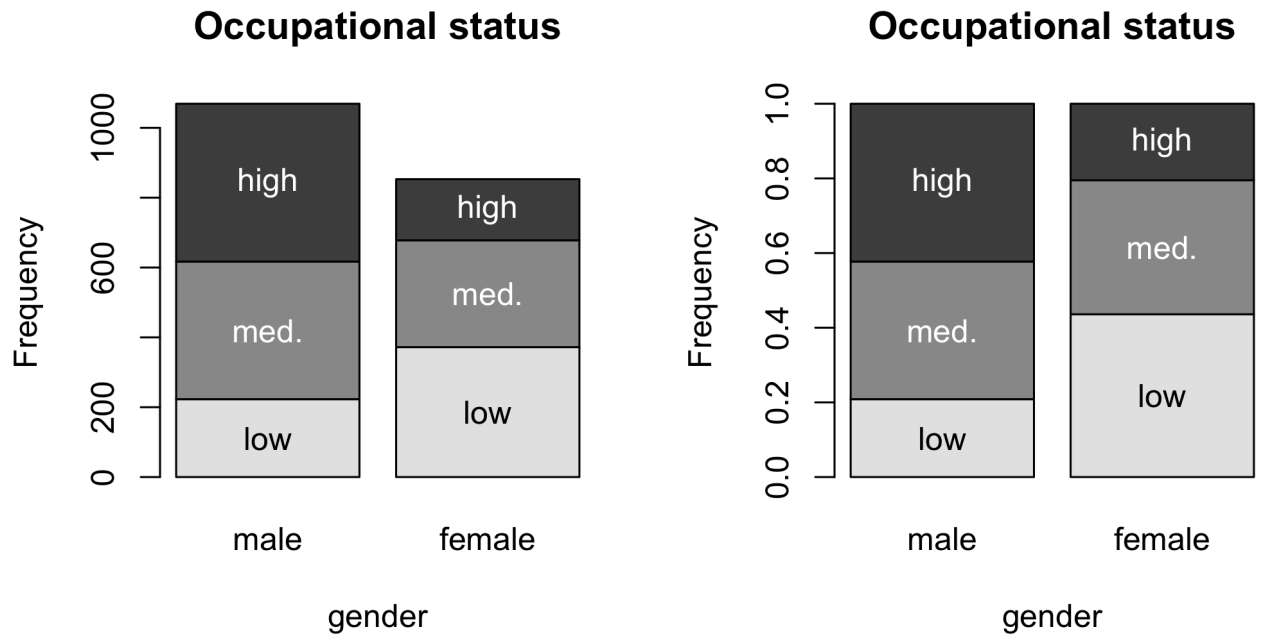






Visualization of a Discrete Variable's Frequency Distribution

Stacked barplot: Frequencies & Proportions



Barplot & Pie Chart: R Syntax

```
tab <- table(occ, sex)
ptab <- prop.table(tab, 2)

tab
```

```
##        sex
## occ     male female
##   low    223    372
##   med.   394    306
##   high   452    175
```

```
ptab
```

```
##        sex
## occ          male    female
##   low  0.2086062 0.4361079
##   med. 0.3685688 0.3587339
##   high 0.4228251 0.2051583
```

Barplot & Pie Chart: R Syntax

```
par(mfcol = c(2,2)) # multiple frames (2 rows, 2 cols)
barplot(tab[, 1], xlab = 'Occ...', ylab = 'Frequency',
        col = c('grey90', 'grey60', 'grey30'), main = 'Men')
barplot(tab[, 2], xlab = 'Occ...', ylab = 'Frequency',
        col = c('grey90', 'grey60', 'grey30'), main = 'Women')
pie(tab[, 1], col = c('grey90', 'grey60', 'grey30'), main = 'Men')
pie(tab[, 2], col = c('grey90', 'grey60', 'grey30'), main = 'Women')

par(mfrow = c(1, 2)) # multiple frames (1 row, 2 cols)
barplot(tab, xlab = 'gender', ylab = 'Frequency',
        col = c('grey90', 'grey60', 'grey30'), main ='Occ...')
barplot(ptab, xlab = 'gender', ylab = 'Frequency',
        col = c('grey90', 'grey60', 'grey30'), main ='Occ...')
```

Continuous Variable

Age: observations Y_1, Y_2, ..., Y_{1922}

62 32 56 63 20 38 39 53 49 54 51 52 25 58 40 59 31 48 34 58 26 35 34 43
49 39 18 51 28 51 38 57 57 30 53 59 36 45 57 55 37 56 60 51 38 44 25 30
37 36 24 34 20 61 45 45 23 42 45 25 28 35 29 57 62 51 27 48 36 55 45 62
… … …
33 54 45 30 33 33 30 56 58 35 28 64 45 24 60 32 41 58 53 31 39 40 45 23
34 28 50 62 26 56 41 41 32 52 48 30 53 55 29 46 45 33 35 31 28 25 33 35
59 57

Age: ordered observations $Y_{(1)}$, $Y_{(2)}$, ..., $Y_{(1922)}$

18 18 18 18 18 18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 21 21 21 21 21 21 21 21 21 21 21 21 21
21 21 21 21 21 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
… … …
62 62 62 62 62 62 62 62 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63 63
63 63 63 64 64 64 64 64 64 64 64 64 64 64 64 64 64 64 65 65 65 65 65 65
65 65

Frequency Table / Distributionof a Continuous Variable

Frequency distribution of age : Table

Age	*f*	*p*	*cf*	*cp*
18	12	0.006	12	0.006
19	8	0.004	20	0.010
...	...	...	...	...
30	50	0.026	448	0.233
31	44	0.023	492	0.256
32	52	0.027	544	0.283
33	43	0.022	587	0.305
34	50	0.026	637	0.331
35	48	0.025	685	0.356
36	44	0.023	729	0.379
37	53	0.028	782	0.407
38	30	0.016	812	0.422
39	48	0.025	860	0.447
40	46	0.024	906	0.471
41	48	0.025	954	0.496
42	53	0.028	1007	0.524
43	43	0.022	1050	0.546
44	34	0.018	1084	0.564
45	49	0.025	1133	0.589
46	42	0.022	1175	0.611
47	48	0.025	1223	0.636
48	51	0.027	1274	0.663
49	48	0.025	1322	0.688
50	41	0.021	1363	0.709
51	53	0.028	1416	0.737
52	49	0.025	1465	0.762
...	...	...	...	...
64	15	0.008	1914	0.996
65	8	0.004	1922	1.000
Total	1922	1.000		

Frequency Table / Distributionof a Discrete Variable

f_i ... frequencies for the i=1,...,K categories

with $f_1 + f_2 + \ldots + f_k = \sum_{i=1}^{k} f_i = N$

p_i ... proportion of cases in the ith category:

$p_i = \frac{f_i}{N}$ with $\sum_{i=1}^{k} p_i = 1$

cf_i ... cumulative frequencies: $cf_i = \sum_{h=1}^{i} f_h$

cp_i ... cumulative proportions: $cp_i = \frac{1}{N}\sum_{h=1}^{i} f_h = \frac{cf_i}{N}$

Cumulative frequencies and proportions are only meaningful for ordered or continuous data!

Frequency Distributionof a Continuous Variable

Frequency distribution of age : Strip chart

Frequency Table: Grouped Data

Frequency table/distribution of grouped age

10 age groups (5 years)

Age	*f*	*p*	*cf*	*cp*
16-20	35	0.018	35	0.018
21-25	178	0.093	213	0.111
26-30	235	0.122	448	0.233
31-35	237	0.123	685	0.356
36-40	221	0.115	906	0.471
41-45	227	0.118	1133	0.589
46-50	230	0.120	1363	0.709
51-55	233	0.121	1596	0.830
56-60	224	0.117	1820	0.947
61-65	102	0.053	1922	1.000
Total	1922	1.000		

5 age groups (10 years)

Age	*f*	*p*	*cf*	*cp*
16-25	213	0.111	213	0.111
26-35	472	0.246	685	0.356
36-45	448	0.233	1133	0.589
46-55	463	0.241	1596	0.830
56-65	326	0.170	1922	1.000
Total	1922	1.000		

Frequency Distribution: Grouped Data

Frequency

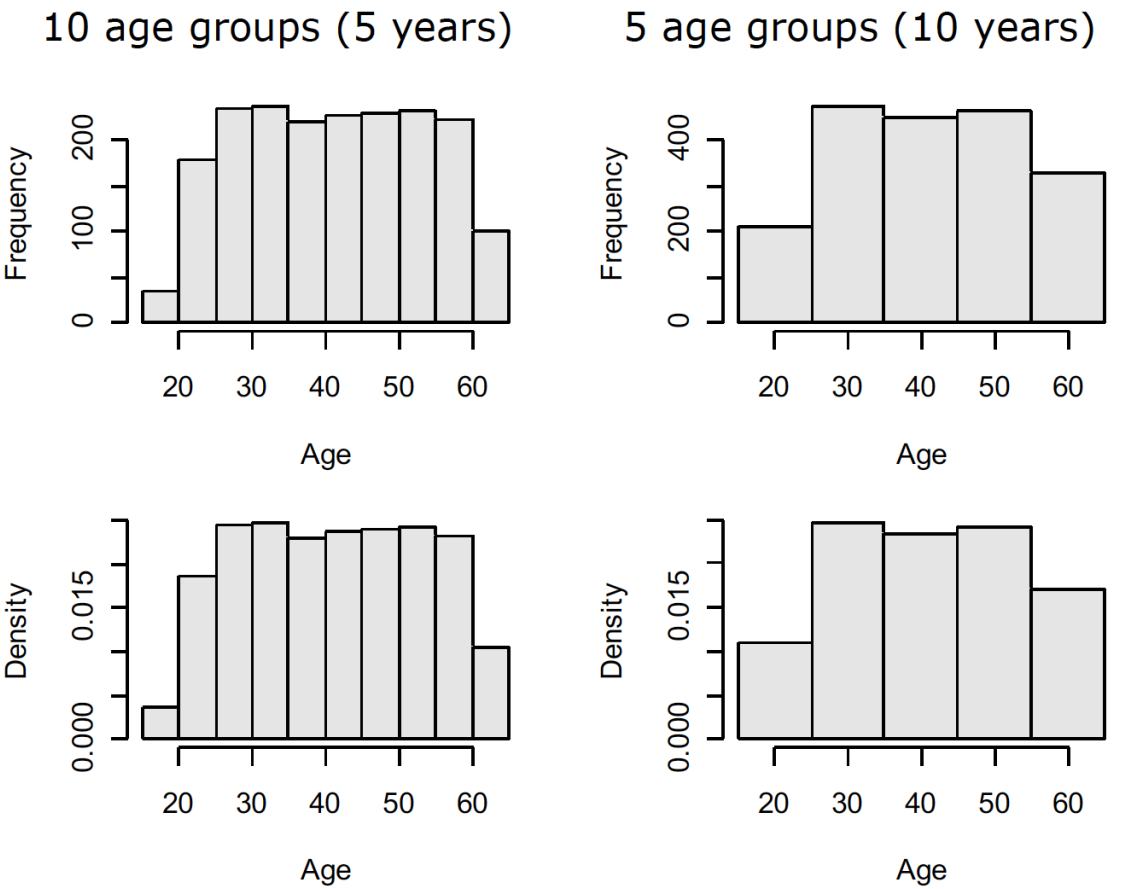
Density:

$d_i = p_i / w_i$

w_i ... width of group i

Area of histogram =1

$\sum d_i \cdot w_i = 1$

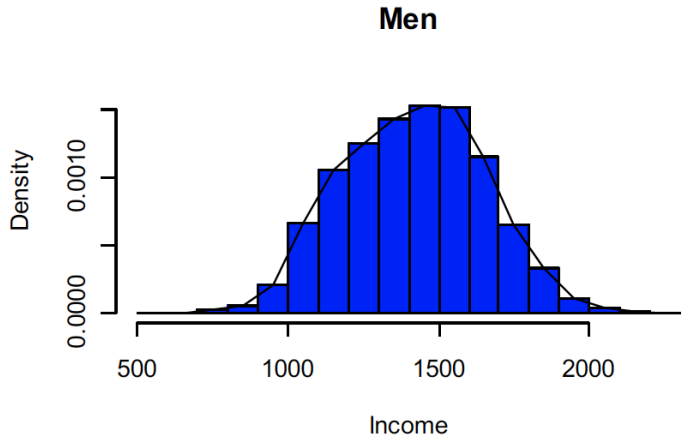


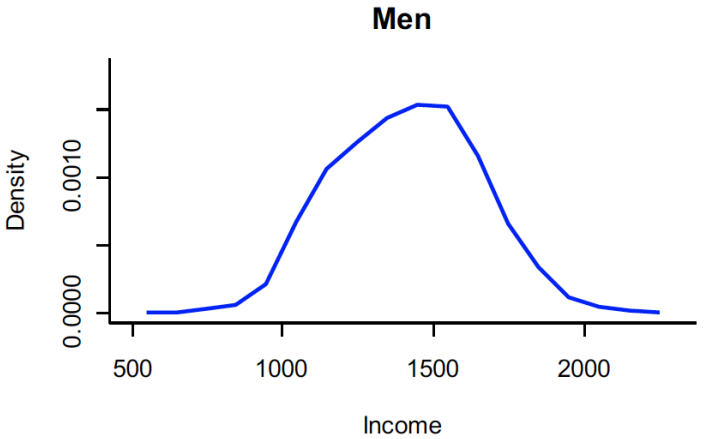
Histogram: R Syntax

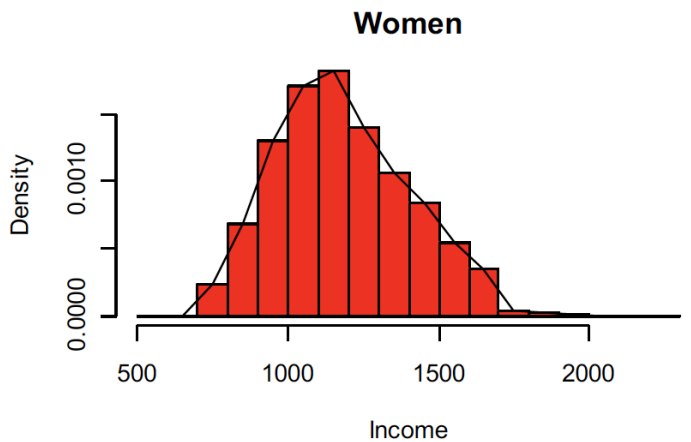
```
par(mfrow = c(2,2))
# histogram with frequencies
hist(age, seq(15, 65, by = 5), xlab = 'Age',
     main = '', col = 'grey90')
hist(age, seq(15, 65, by = 10), xlab = 'Age',
     main = '', col = 'grey90')

# histogram with density
hist(age, seq(15, 65, by = 5), xlab = 'Age',
     main = '', col = 'grey90', prob = T)
hist(age, seq(15, 65, by = 10), xlab = 'Age',
     main = '', col = 'grey90', prob = T)
```

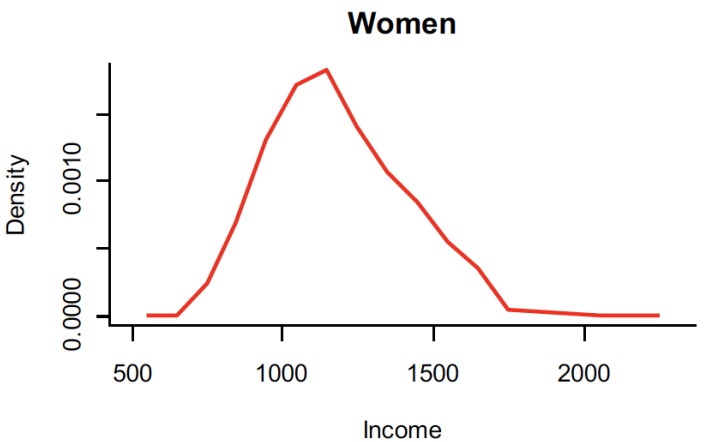
- `seq()` is used for breakpoints of histogram.
- `prob = T` results in plotting the density instead of frequency.

Histogram & Polygon (Income)

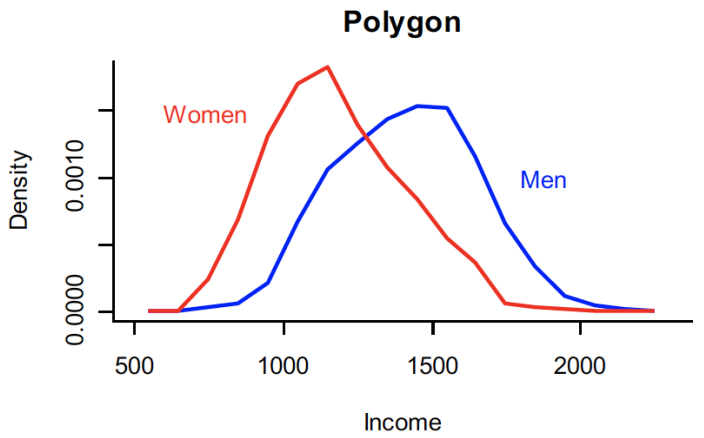


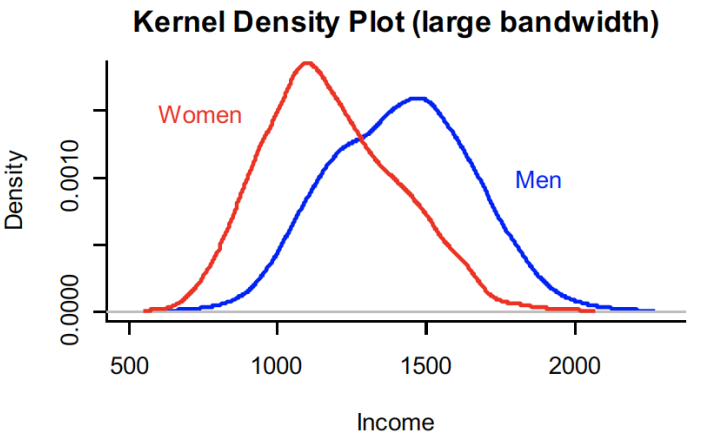


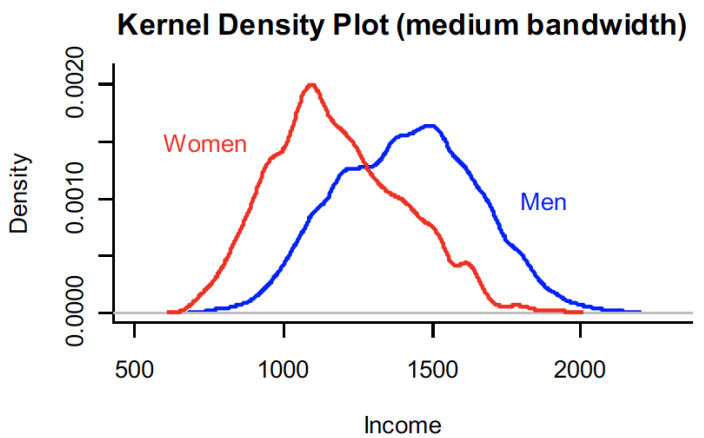


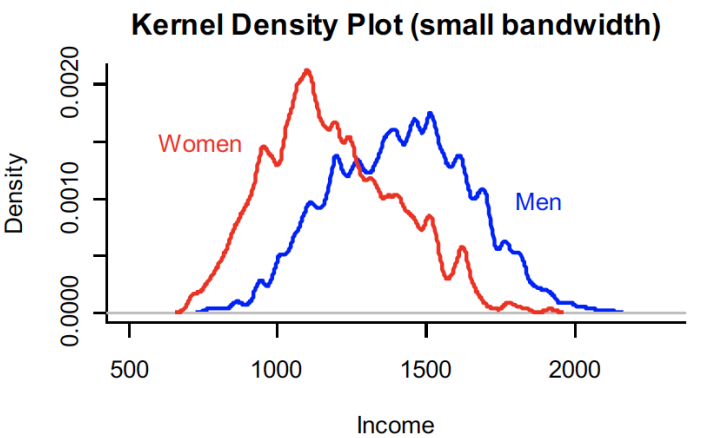


Polygon & Kernel Density Plots









Kernel Density Estimation

Nonparametric density estimation, i.e., we are not assuming any distributional shape like a normal distribution.

In order to get a density estimate for a specific point x we only look at a very close symmetric neighborhood of x (similar to a histogram bin but centered at x). In addition, we weight observations X_i :

- the closer to x the more weight observation X_i gets
- weights are determined by the kernel function $K(z)$; the kernel function might be a normal, triangular, rectangular, or other distribution (tricube and Epanechnikov kernels are rather popular)

Kernel Density Estimation

More formally, the kernel density estimator at a specific point x is given by

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h})$$

- X_i ... observation i, for $i = 1, \ldots, n$
- $K(z)$ ... kernel density
- h ... bandwidth
- n ... sample size

Examples for kernels :

- Normal (Gaussian) kernel: $K(z) = (1/\sqrt{2\pi})e^{-z^2/2}$
- Rectangular (uniform) kernel:

$$K(z) = \begin{cases} 1/2 & \text{for } |z| < 1 \\ 0 & \text{for } |z| \geq 1 \end{cases}$$

Kernel Density Estimation

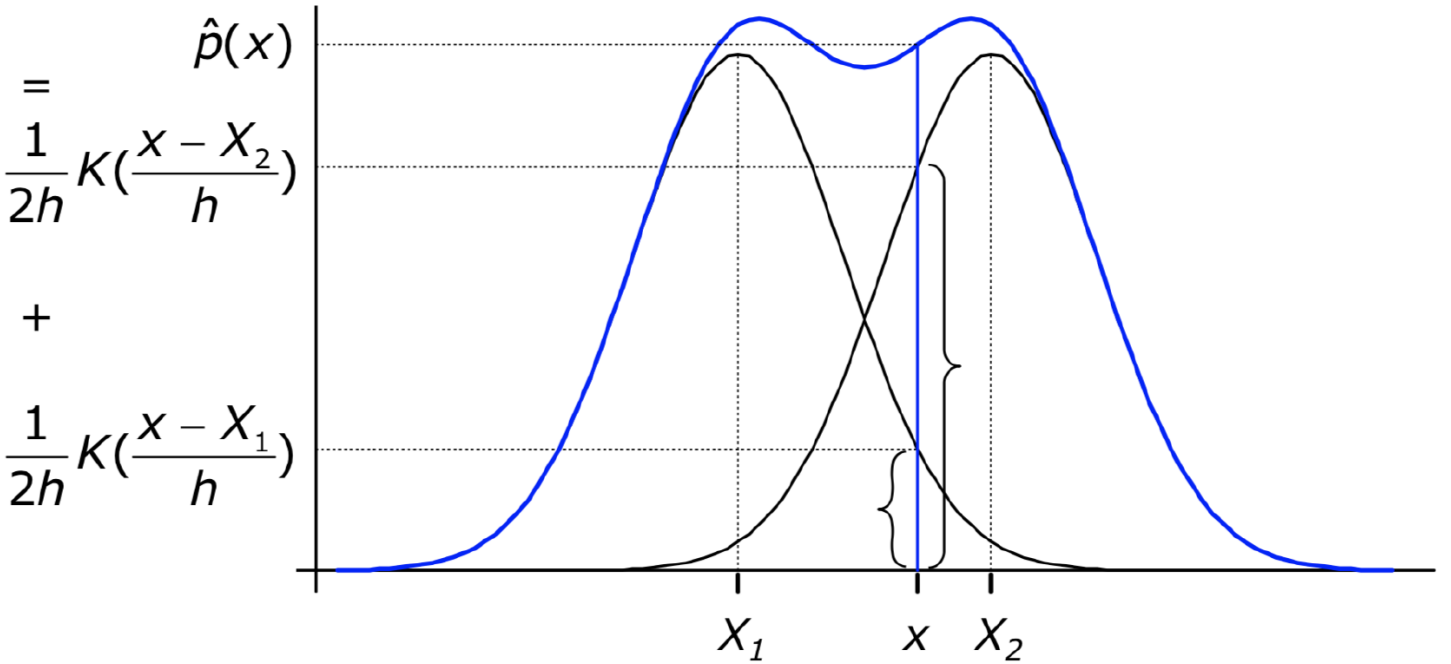
How to read the formula:

$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h})$$

- $\frac{1}{nh}$: Scaling factor; scales the area under the density function to 1 (=definition of a density)
- $\sum_{i=1}^{n} K(\frac{x-X_i}{h})$: “Counts” the number of observations in the close neighborhood of x. More precisely, it is the sum of kernel weights.
  - the farther away observation X_i from x, the less weight
  - the larger the bandwidth h (=neighborhood), the more weight distant observations get

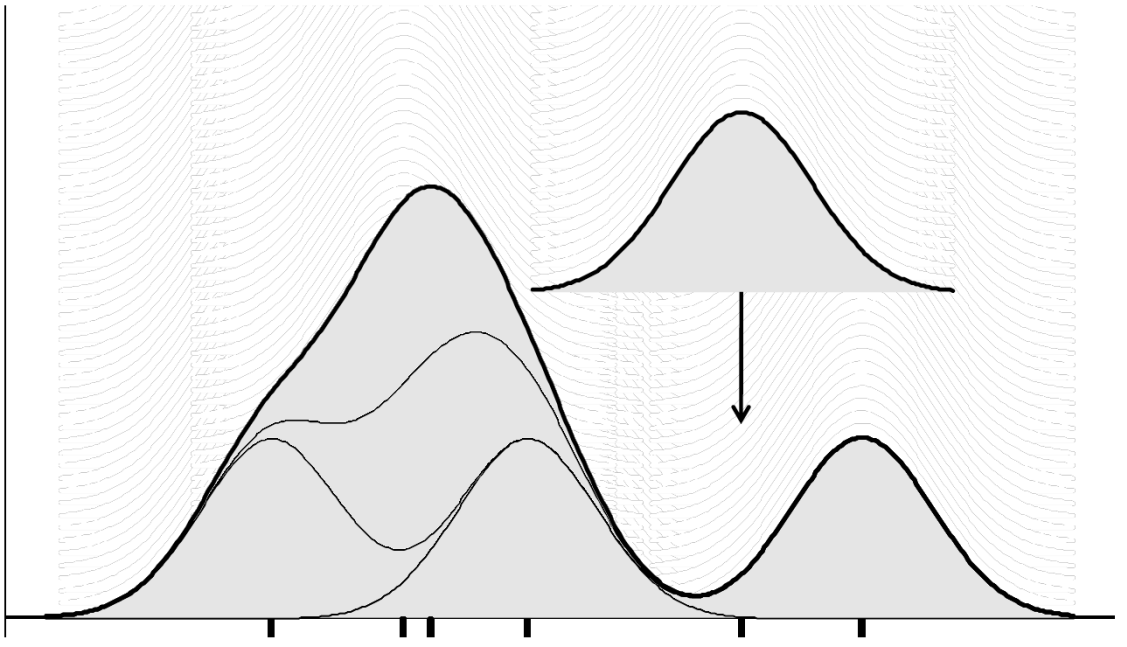
Kernel Density Estimation

Simple example with two observations X_1, X_2 and a normal kernel

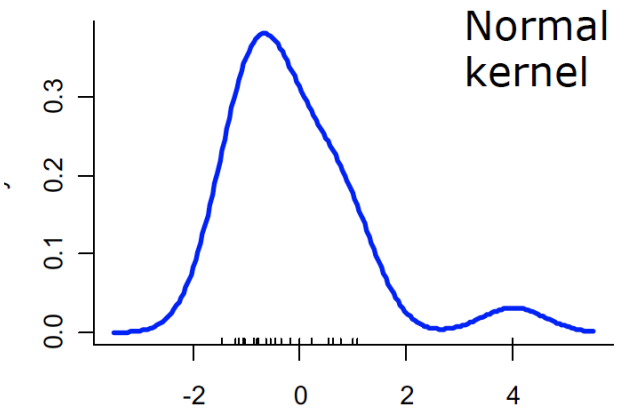


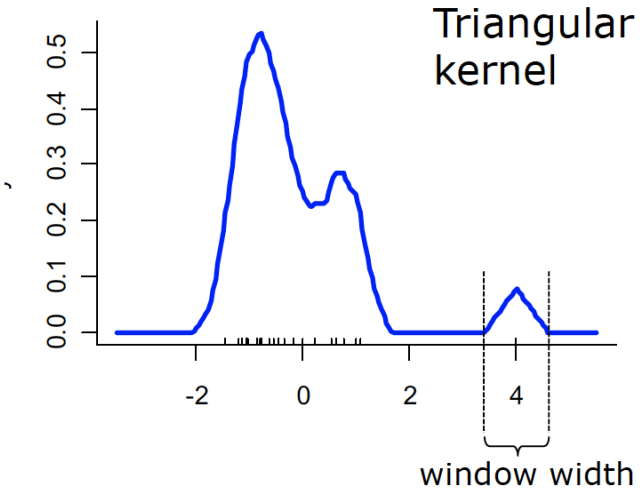
Kernel Density Estimation

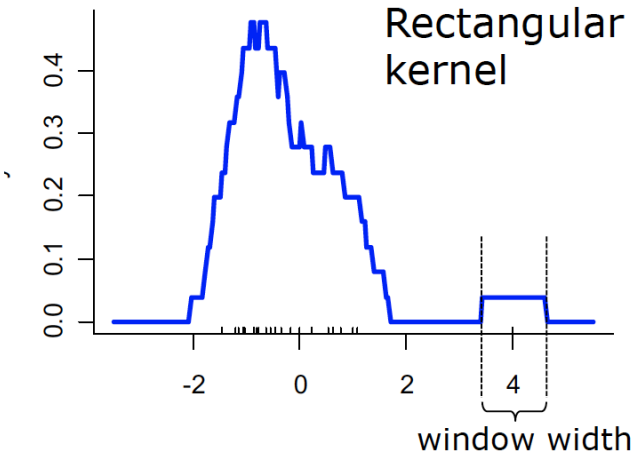
Alternative point of view: “drop” a kernel for each observation



Kernel Density Estimation





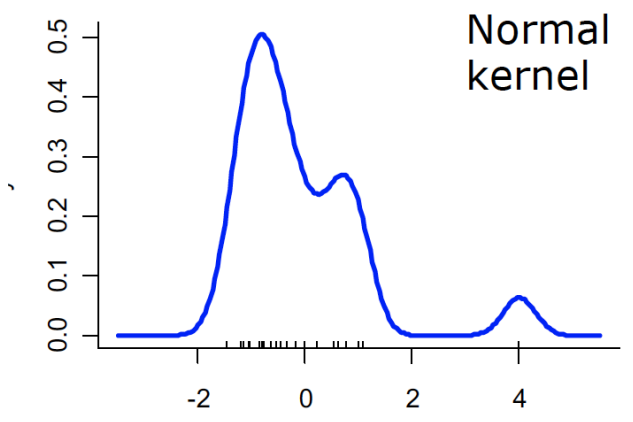


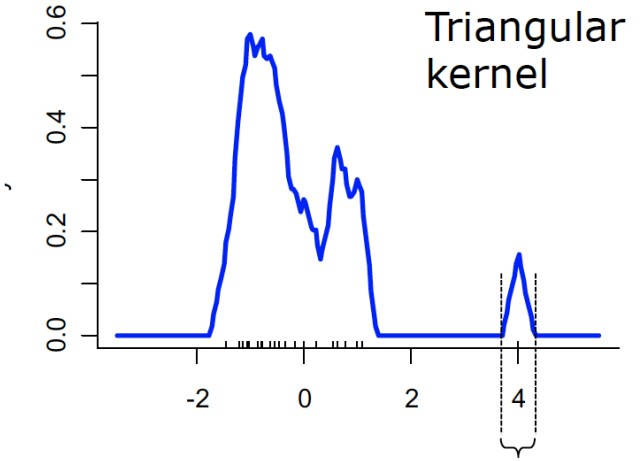
Bandwidth: *h* = 0.6

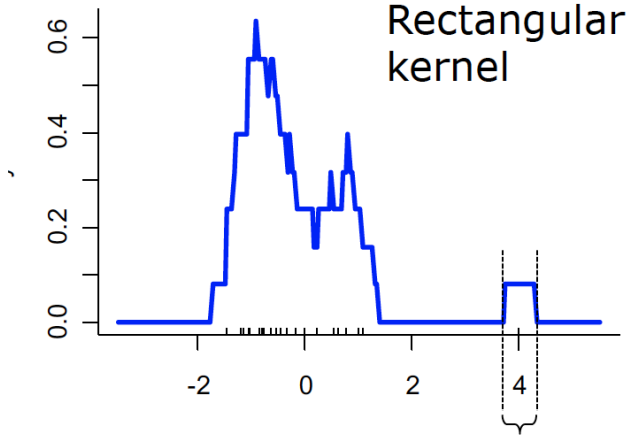
Normal:	stddev = .6
Triangular:	window = 2 x .6
Rectangular:	window = 2 x .6

(window width = 2 x bandwidth)

Kernel Density Estimation





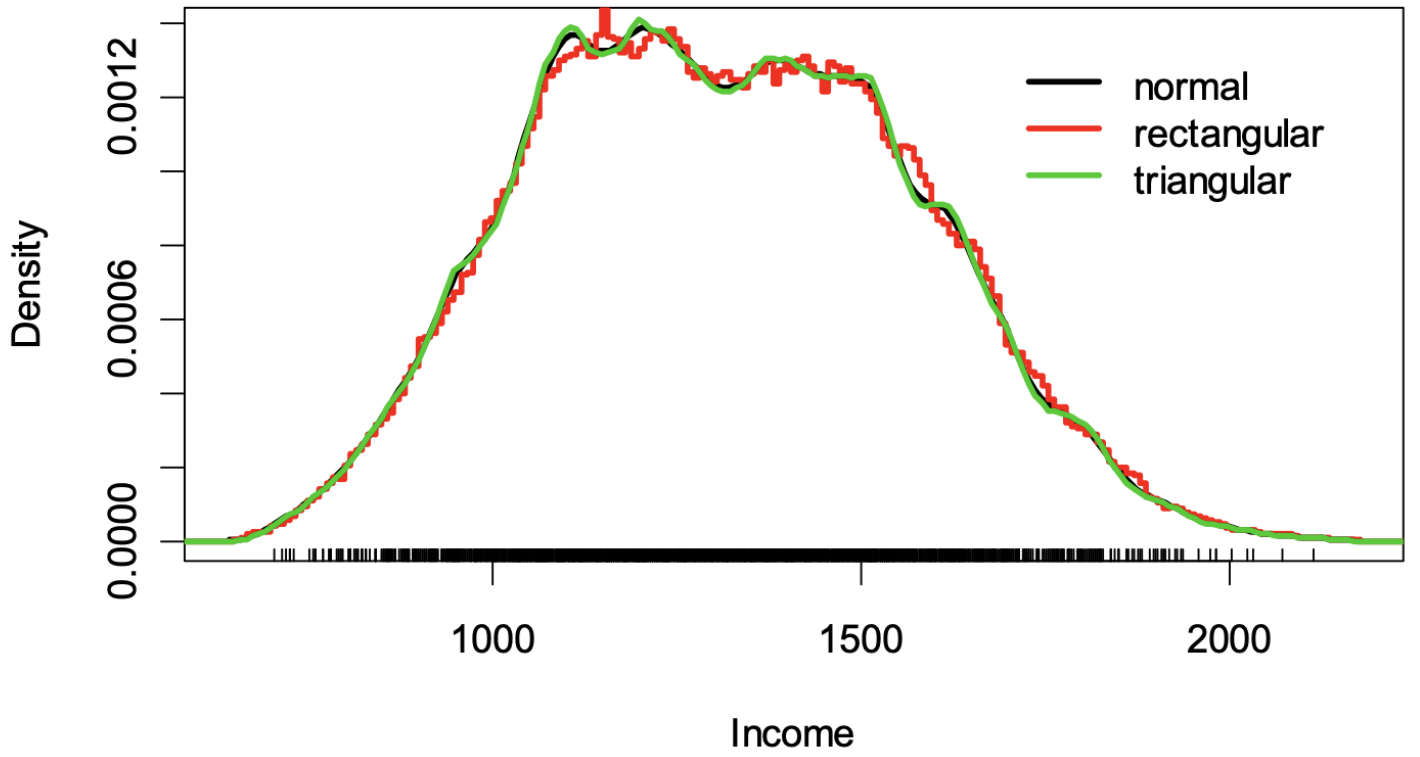


Bandwidth: *h* = 0.3

Normal:	stddev = .3
Triangular:	window = 2 x .3
Rectangular:	window = 2 x .3

Kernel Density Estimation

Example: Income distribution



How to Do it in R?

Write your own R-function

```
kden <- function(x, h, x.val = seq(min(x)-2*h,
max(x)+2*h, length = 200))
{
  x.val <- sort(unique(x.val))  # sort values
  n <- length(x)                # number of obs.
  dens.f <- function(x.v) {     # kernel density est.
  z <- (x.v - x) / h         # rectangular kernel
  d <- ifelse(z > -1 & z < 1, .5, 0)
  sum(d) / n / h
}

  dens <- sapply(x.val, dens.f) # apply to each x.val
  data.frame(x = x.val, y = dens)
}
```

How to Do it in R?

```
# plot output of kden()
plot(kden(incex$income, h = 50), main = 'Income Distribution')
```

Use available R-function: `density()`

- `density()` slightly differs from `kden()` due to automatic bandwidth determination

```
plot(density(incex$income), main = 'Income Distribution')
```

Quantiles

- Ordered observations $Y_{(1)}, Y_{(2)}, \ldots, Y_{(N)}$ are partitioned into q groups of equal size. The (observed) values which separate the q groups are called quantiles.
- **Quartiles**: $q = 4$ equally sized groups consisting of 25% of observations each
  - $Q_1 = Y_{.25}$: The first quartile is the smallest observation for which holds that 25% of all observations are smaller or equal to it.
  - $Q_2 = Y_{.50}$: The second quartile is the smallest observation for which holds that 50% of all observations are smaller or equal to it.
  - $Q_3 = Y_{.75}$: The third quartile is the smallest observation for which holds that 75% of all observations are smaller or equal to it.

Quantiles

The construction principle is similar for other quantiles, e.g.,

- **Quintiles**: $Y_{.2}, Y_{.4}, Y_{.6}$, and $Y_{.8}$ partition all observations into $q = 5$ equally sized groups consisting of 20% of observations each.
- **Deciles**: $Y_{.1}, Y_{.2}, \ldots, Y_{.8}, Y_{.9}$ partition all observations into $q = 10$ equally sized groups consisting of 10% of observations each.
- **Percentiles**: more generally, Y_p is the p percentile; Y_p is the smallest value for which holds that at least p of observations are smaller than or equal to Y_p.

Quantiles (Example)

Quartiles of the *age* distribution

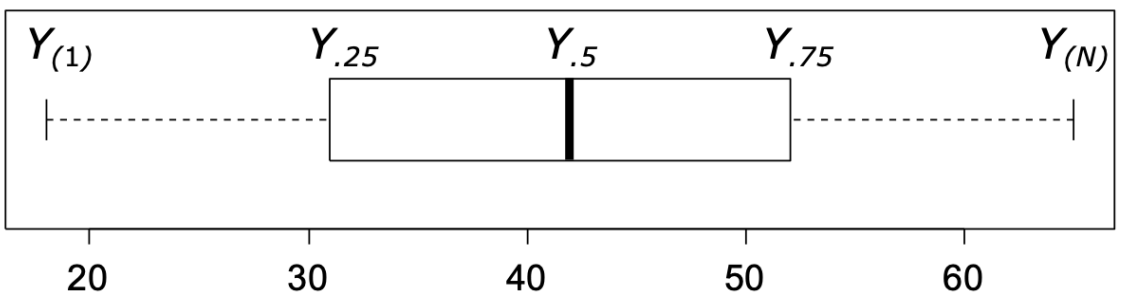
Age	*f*	*p*	*cf*	*cp*
18	12	0.006	12	0.006
19	8	0.004	20	0.010
...	...	...	...	...
30	50	***first quartile***		233
$Y_{.25}$ 31				0.256
32	52	0.027	544	0.283
33	43	0.022	587	0.305
34	50	0.026	637	0.331
35	48	0.025	685	0.356
36	44	0.023	729	0.379
37	53	0.028	782	0.407
38	30	0.016	812	0.422
39	48	0.025	860	0.447
40	46	0.024	906	0.471
41	48	0.025	954	0.496

Age		***second quartile***		*cp*
$Y_{.5}$ 42				0.524
43	43	0.022	1050	0.546
44	34	0.018	1084	0.564
45	49	0.025	1133	0.589
46	42	0.022	1175	0.611
47	48	0.025	1223	0.636
48	51	0.027	1274	0.663
49	48	0.025	1322	0.688
50	41	0.021	1363	0.709
51		***third quartile***		0.737
$Y_{.75}$ 52				0.762
...	...	...	...	...
64	15	0.008	1914	0.996
65	8	0.004	1922	1.000
Total	1922	1.000		

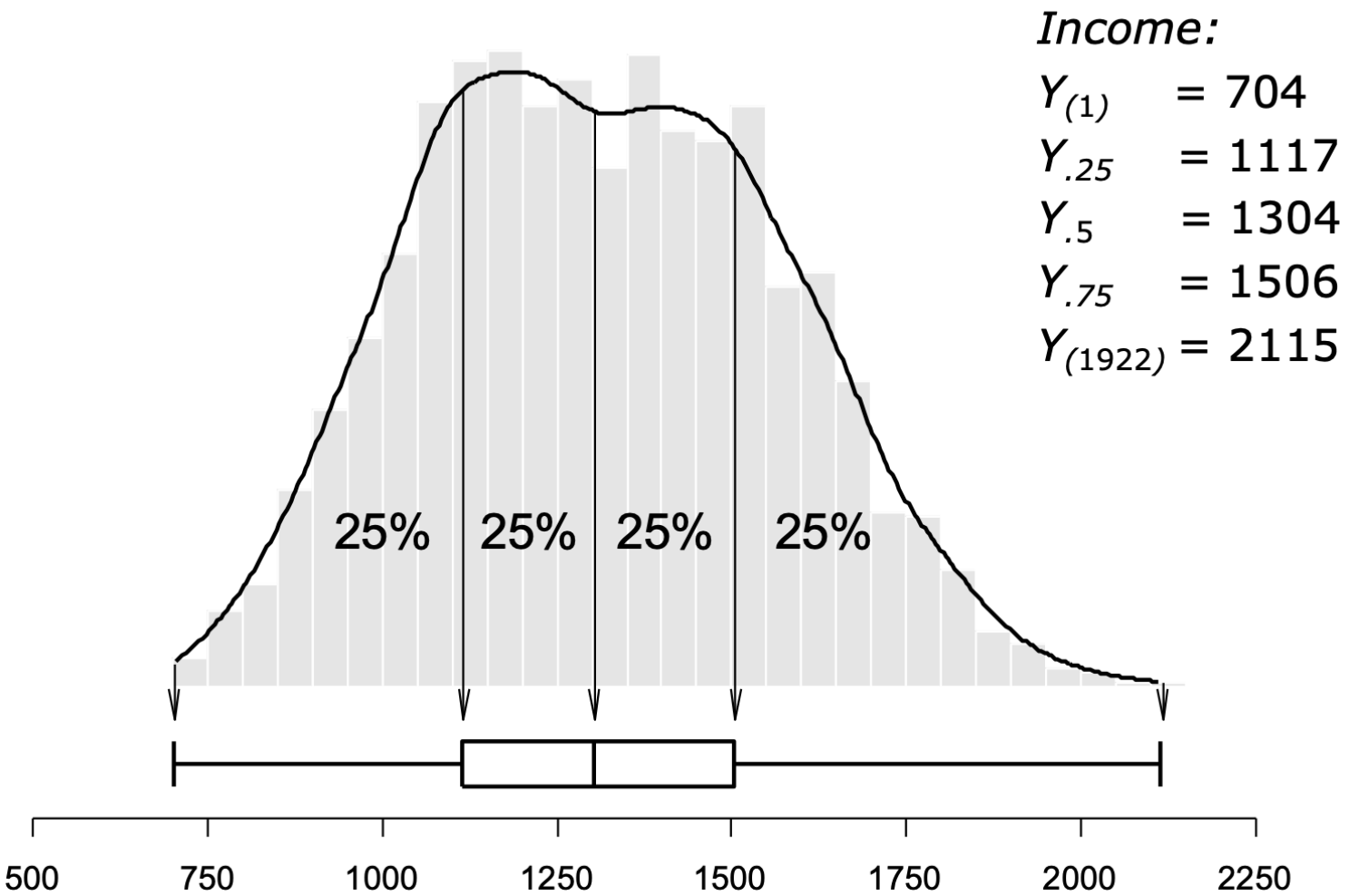
Visualization of Quartiles: Boxplot

The boxplot is a visual display of five statistics:

	statistic	e.g., *age*
– Minimum of Y ($min(Y)$):	$Y_{(1)}$	18
– 1. quartile of Y:	$Y_{.25}$	31
– 2. quartile of Y (median):	$Y_{.5}$	42
– 3. quartile of Y:	$Y_{.75}$	52
– Maximum of Y ($max(Y)$):	$Y_{(N)}$	65



Visualization of Quartiles: Boxplot

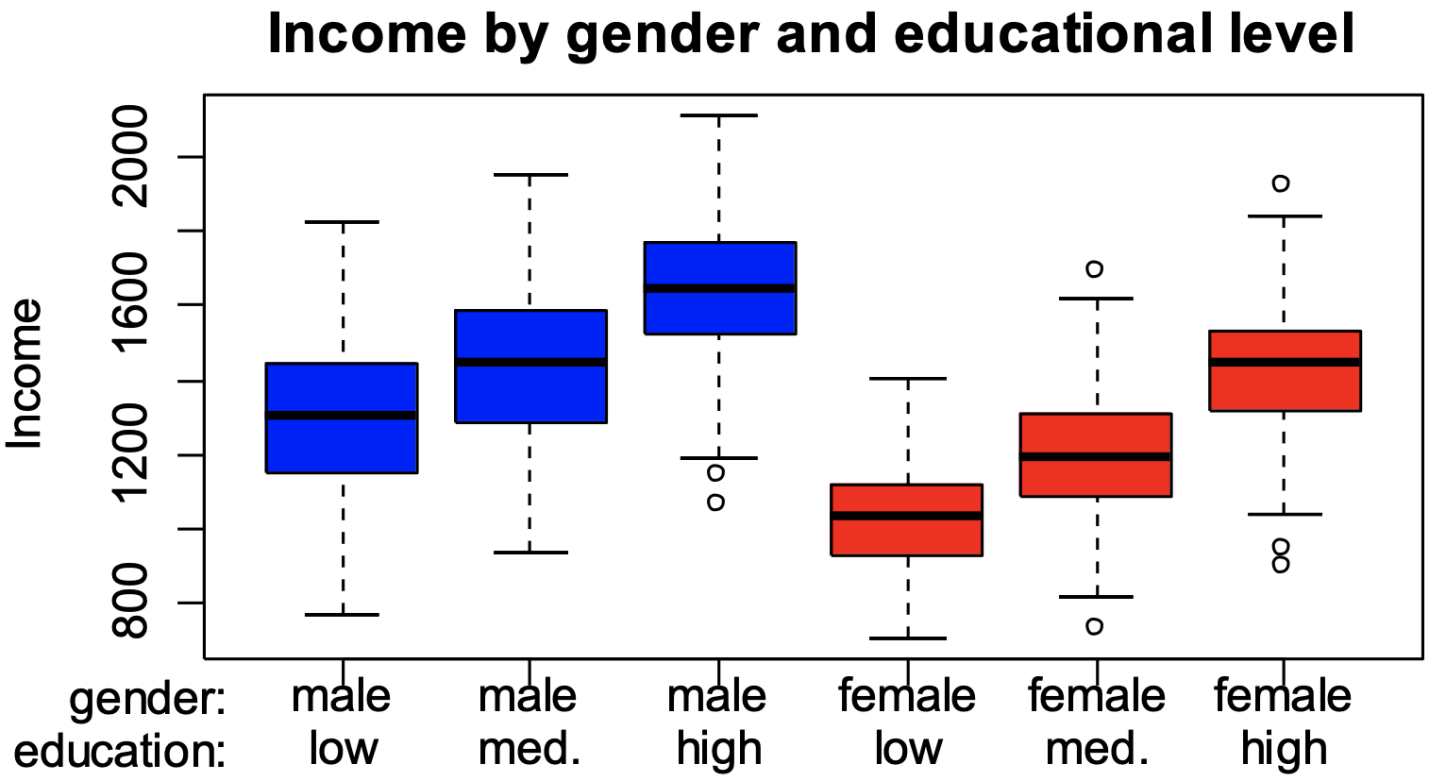


Quantiles & Boxplots

There are different ways (formulas) for computing quantiles and displaying boxplots. Different statistical software tools use different approaches (frequently the percentile function). However, for large data sets differences in results (quantiles and boxplots) are minor—substantive interpretations do not change!

An alternative version of the boxplot also displays outliers (i.e., extreme observations that stand apart from the rest of the distribution). An outlier is drawn if its distance to the box ($Y_{.25}$ or $Y_{.75}$) is larger than 1.5 times the boxwidth, i.e., $1.5 \cdot (Y_{.75} - Y_{.25})$.

Grouped Boxplots (with outliers)



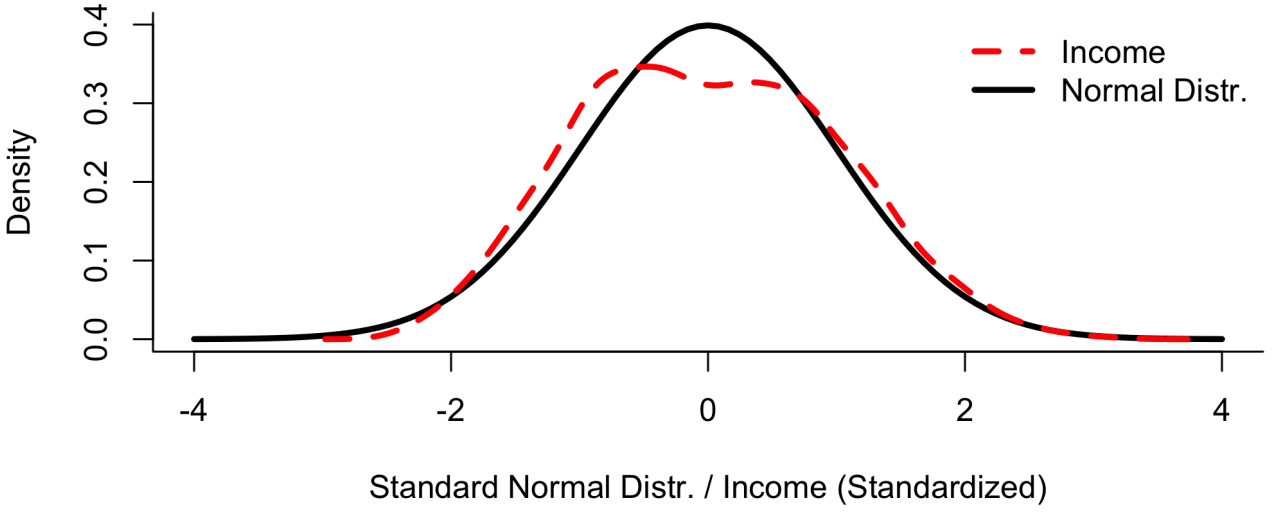
Quantile-Comparison Plot

We want to check whether the observed (empirical) sample distribution of a variable resembles a theoretical distribution (e.g., normal distribution).

We could do so by comparing observed and theoretically expected histograms or kernel density estimates; but the comparison is more efficient using quantile-comparison plots (no bins or bandwidths need to be specified).

Empirical & Theoretical Distributions

```
curve(dnorm, -4, 4, lwd = 3, bty = 'L', ylab = 'Density',
      xlab = 'Standard Normal Distr. / Income (Standardized)')
lines(density(scale(incex$income)), col = 'red', lwd = 3,
      lty = 2)
legend('topright', c('Income', 'Normal Distr.'), bty = 'n',
      col = c('red', 'black'), lty = c(2, 1), lwd = 3)
```



Quantile-Comparison Plot

In order to determine the theoretical quantiles , you need to do the following:

1. Order the data values from smallest to largest: $X_{(1)}, X_{(2)}, \ldots, X_{(n)}$
2. Determine the cumulative proportions of values that are less than $X_{(i)}$:

$$P_i = \frac{i - \frac{1}{2}}{n}$$

- i/n (for $i = 1, \ldots, n$) would be the cumulative probability related to the empirical quantile. Since for the largest observation $n/n = 1$, we would obtain the maximum value of the theoretical distribution, e.g., $+\infty$ for the normal distribution (moreover, theoretical quantiles would not be symmetric). In order to avoid this we subtract ½ from i.

Quantile-Comparison Plot

1. Use the inverse of the theoretical cumulative distribution function (CDF) to find the z_i value corresponding to the cumulative probability P_i:
2. Plot z_i (horizontal axis) against $X_{(i)}$ (vertical axis)

- If the plot shows all points on a straight line, the data can be assumed to be sampled from the underlying theoretical distribution.
- Due to sampling variation we expect some departure from the linearity—plotting a 95% confidence "envelope" helps in assessing significant departures from the theoretical distribution

How to Do it in R?

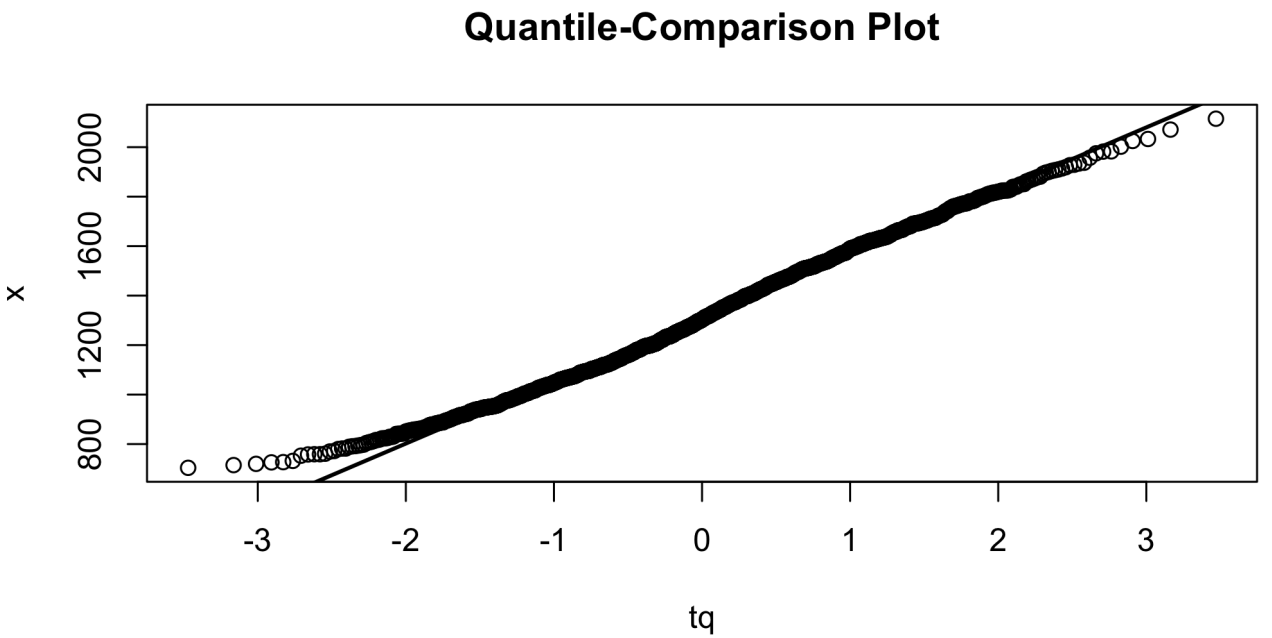
Write your own function qq() :

```
qq <- function(x, ...)
{
  n <- length(x)          # number of observations
  x <- sort(x)            # rank-ordered data
  p <- (1:n - .5) / n     # cumulative proportions
  tq <- qnorm(p)          # theoret. (normal) quantiles
  plot(tq, x, main = 'Quantile-Comparison Plot', ...)
  lines(tq, mean(x) + tq * sd(x), lwd = 2)

  # return data frame but do not print on screen
  # that is achieved by invisible()
  invisible(data.frame(x = x, p = p, tq = tq))
}
```

- "`...`" allows passing further arguments to plot() like `ylab`, `xlab`, `pch`, `cex`.

How to Do it in R?

```
qq(incex$income)
```



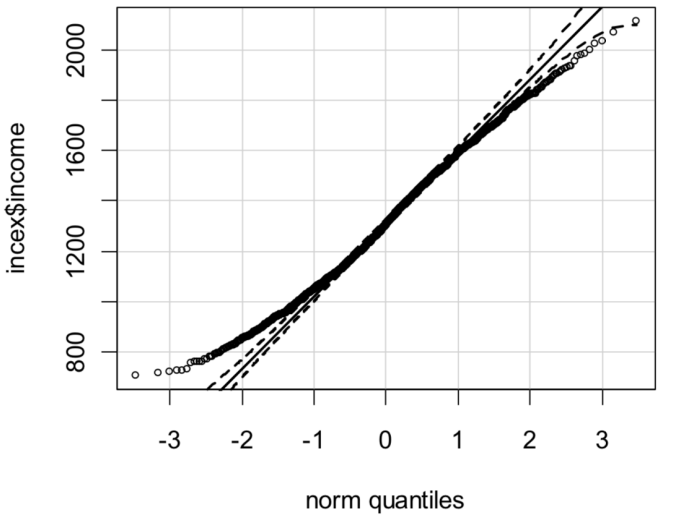
Quantile-Comparison Plot

Use available functions `qqnorm()` or `qqPlot()` :

```
library(car)  # need the car-library for qqPlot()
qqPlot(incex$income, col = 1, cex = .7)
```

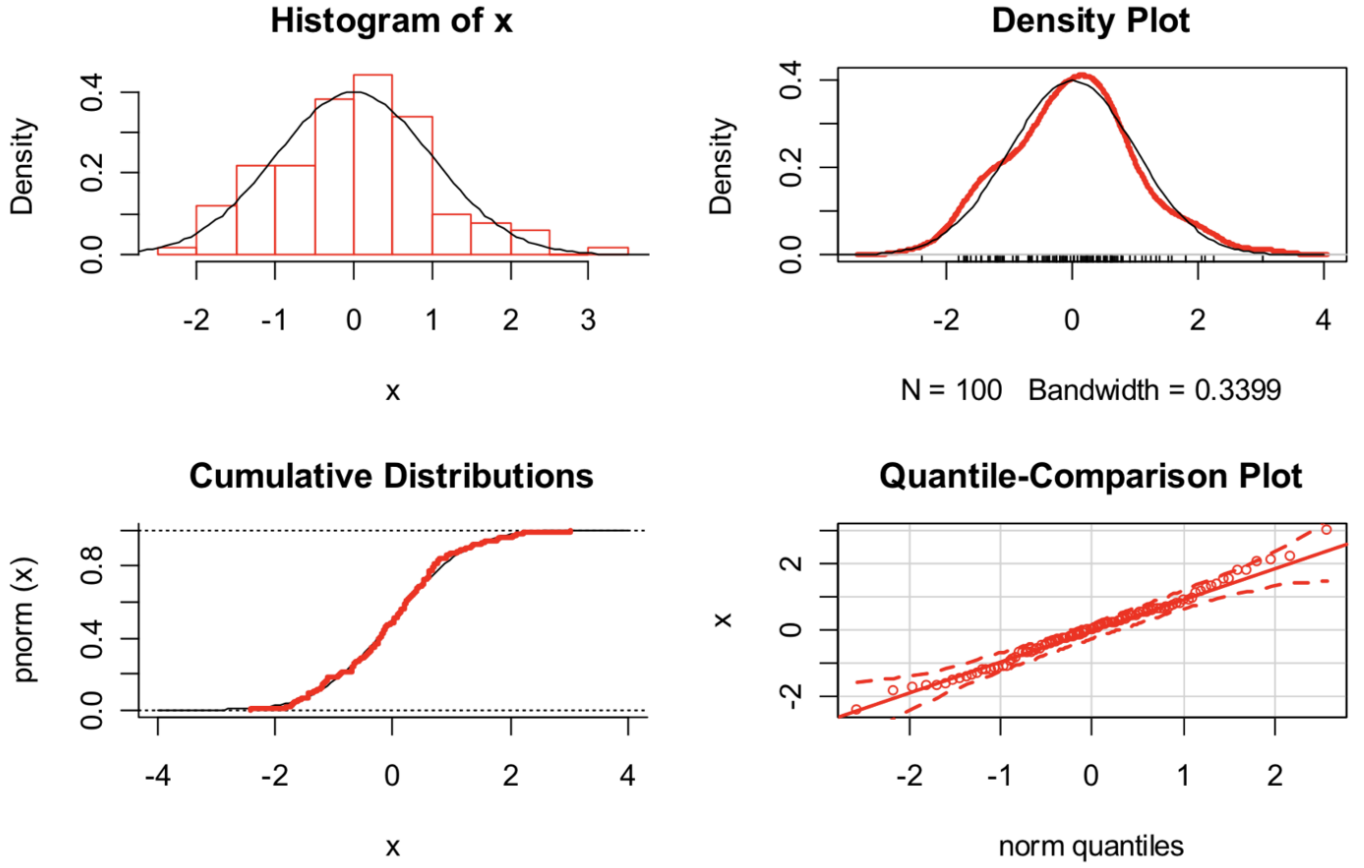
Quantile-comparison plot for *Income* (with normal distribution):

- Clear departure from the theoretically expected line (normal distribution)
- Income is definitely not normally distributed.



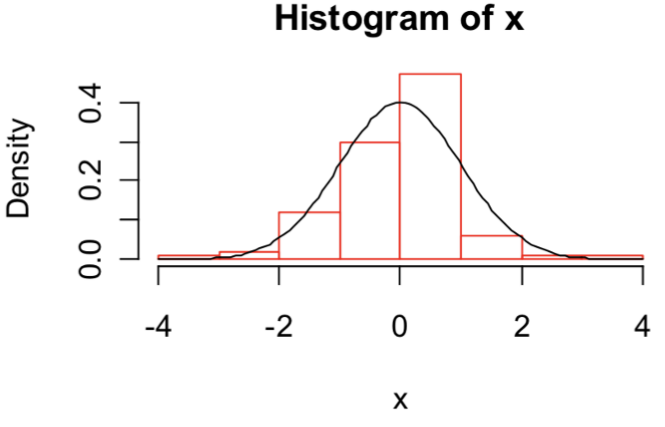
Comparison of Emprical & Theoretical Distributions

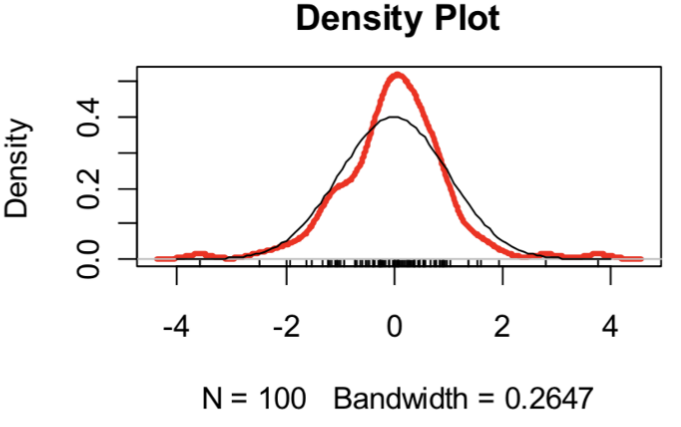
Normally distributed sample data ($n = 100$)

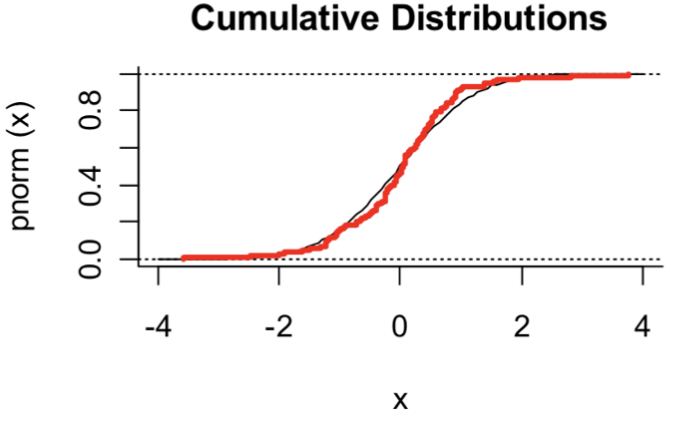


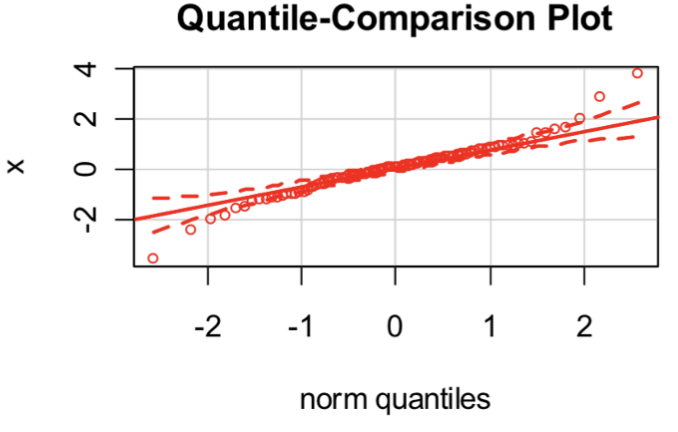
Comparison of Empirical Sample Distribution & Normal Distribution

Sample data with outliers (t -distribution)



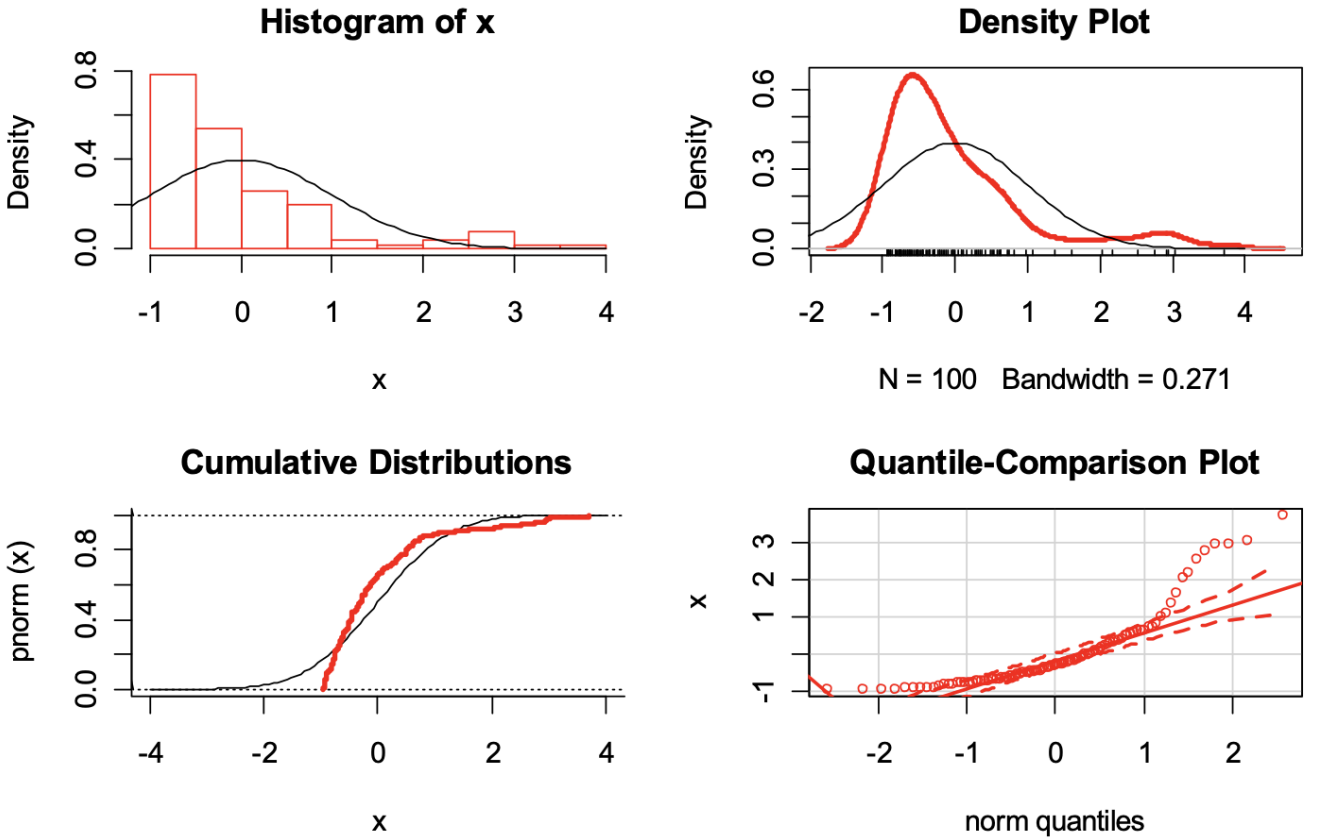






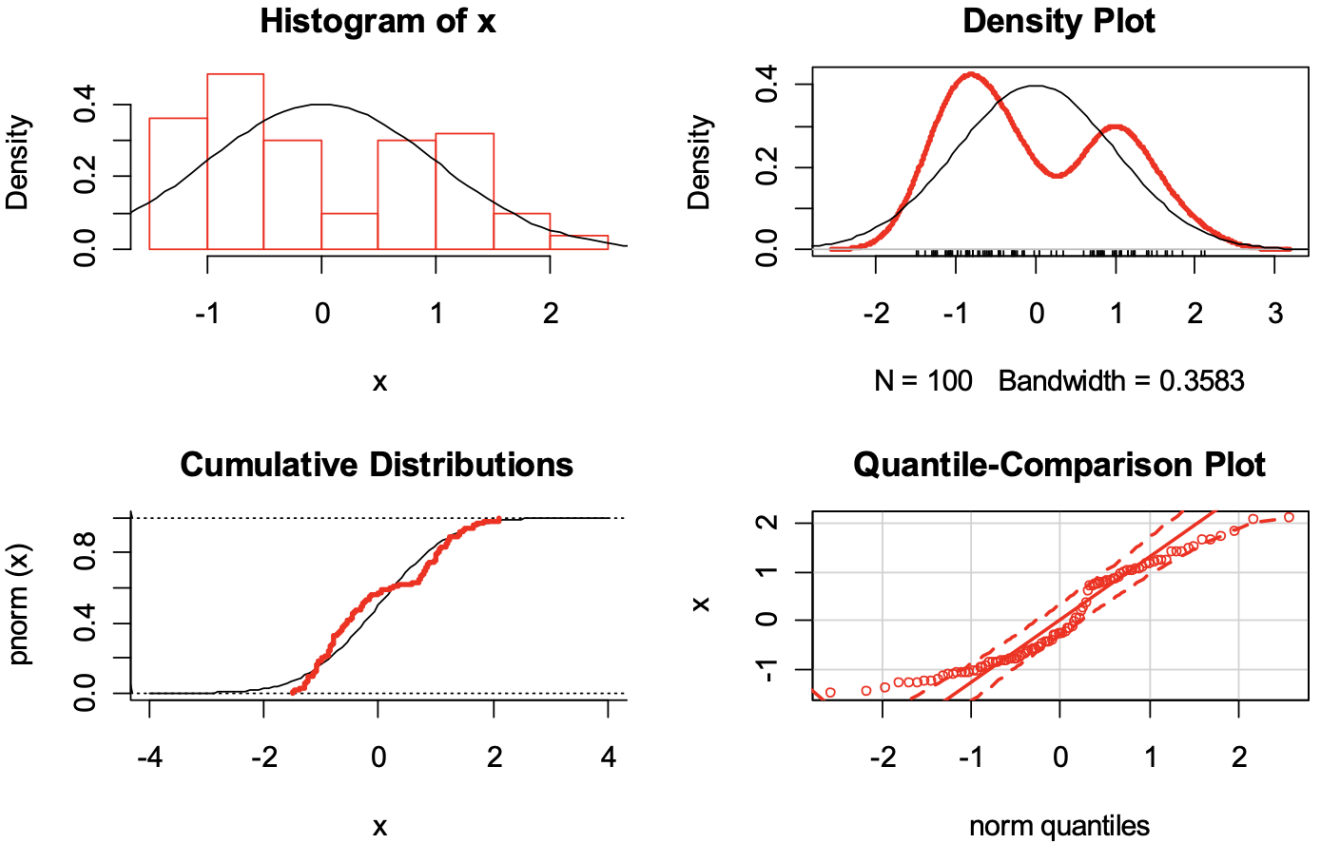
Comparison of Empirical Sample Distribution & Normal Distribution

Right-skewed sample data (chi-square distr.)



Comparison of Empirical Sample Distribution & Normal Distribution

Bimodal sample data (mixture of normal distr.)

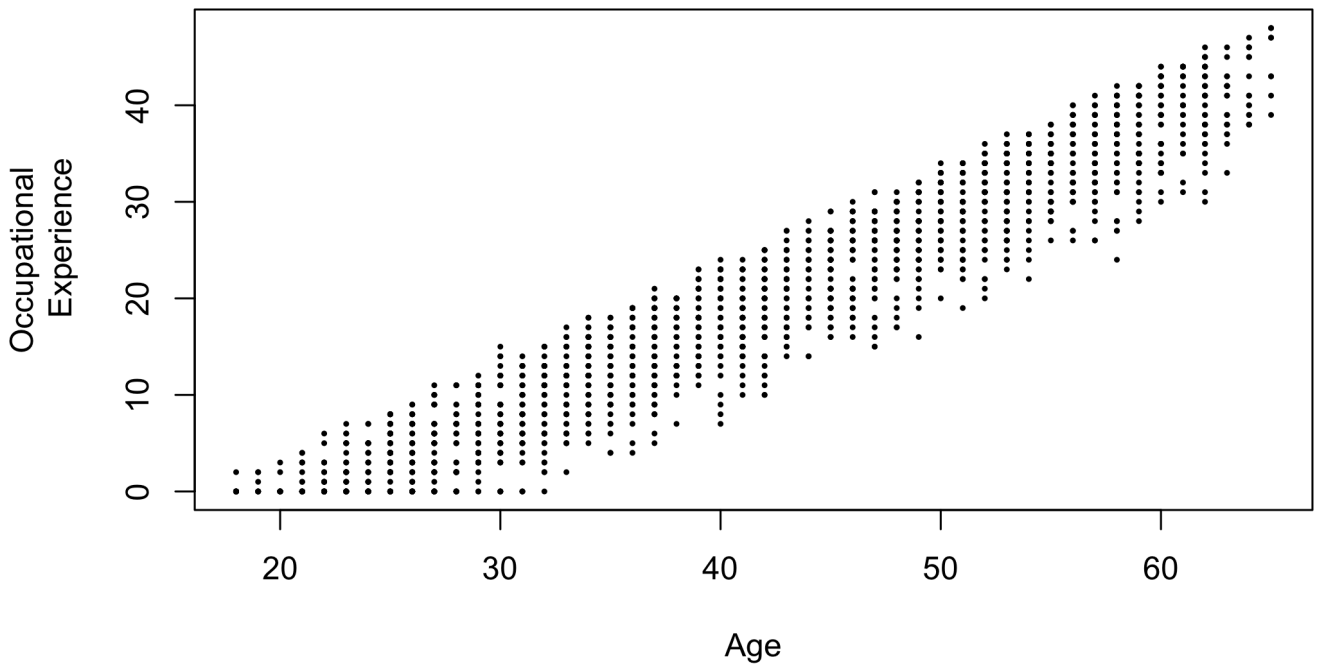


Plotting Bivariate/Multivariate Data

- Scatterplots (jittering & transparency)
    - Continuous variables
    - Categorical variables
- Scatterplot-matrices
- Conditioning plots

Simple Scatterplot

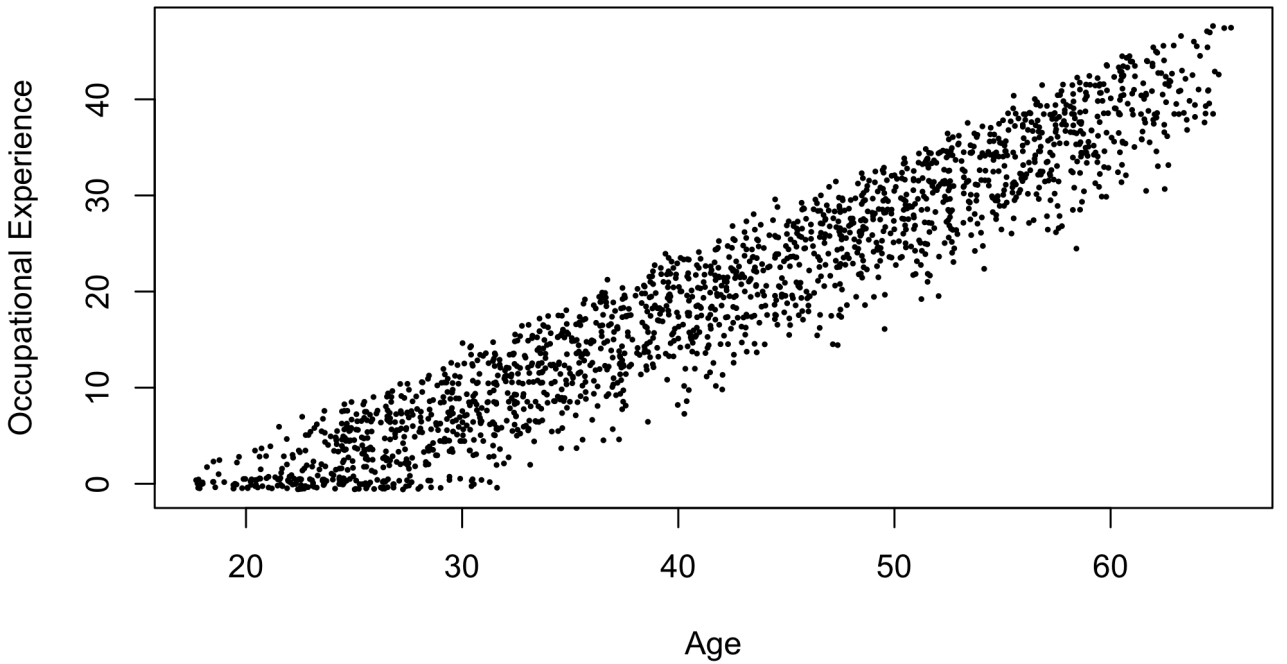
```
par(mar = c(4, 5, 1, 1))
plot(incex$age, incex$oexp, xlab = 'Age', ylab = 'Occupational
     Experience', cex = .4, pch = 16)
```



Note that the bivariate distribution cannot completely be seen, since points overlay (fall on top of each other).

Jittered Scatterplot

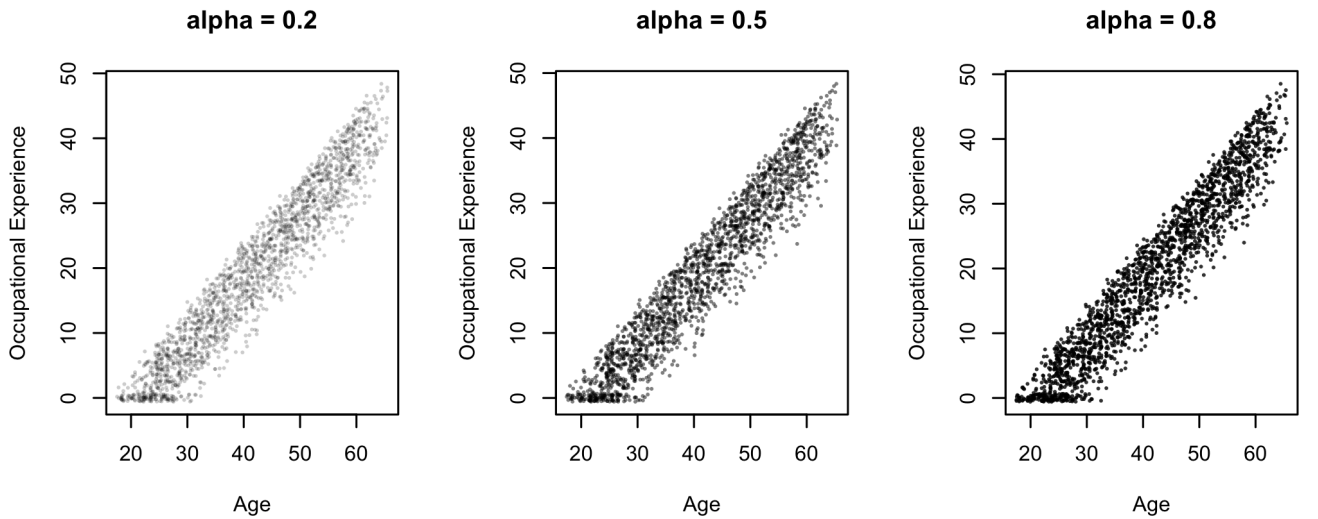
```
par(mar = c(4, 5, 1, 1))
plot(jitter(incex$age, factor = 3), jitter(incex$oexp, factor = 3),
      xlab = 'Age', ylab = 'Occupational Experience', cex = .4, pch =
```



- "jittering" adds a small random quantity (uniformly distributed) to each observation (in that case, for both dimensions)

Scatterplot with different transparency levels

```
par(mfrow=c(1,3))
for (i in c(0.2, 0.5, 0.8)) {
  plot(jitter(incex$age, factor = 3), jitter(incex$oexp, factor = 3),
       xlab = 'Age', ylab = 'Occupational Experience', cex = .4, pch
       col=rgb(red = 0, green = 0, blue = 0, alpha = i), main=paste('
}
```



Scatterplot with different transparency levels

```
par(mfrow=c(1,3))
for (i in c(0.2, 0.5, 0.8)) {
  plot(jitter(incex$age, factor = 3), jitter(incex$oexp, factor = 3),
       xlab = 'Age', ylab = 'Occupational Experience', cex = .4, pch
       col=rgb(red = 0, green = 0, blue = 0, alpha = i), main=paste('
}
```

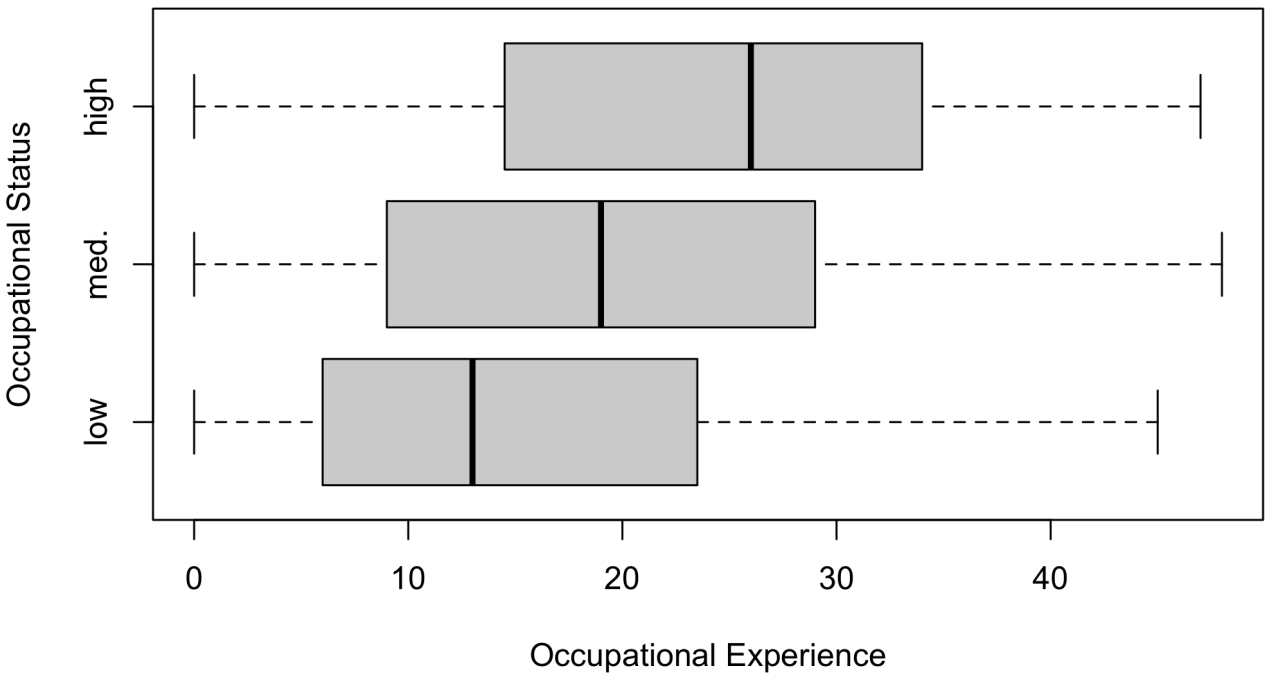
- `alpha` controls the degree of transparency for data points.
- You can install and load the package *scales* (i.e., `library(scales)`) and use the argument `alpha` (e.g., `col=alpha("black", 0.5)`).

Jittered Scatterplot with a Categorical Variable

```
plot(jitter(incex$oexp, factor = 3),
     jitter(as.numeric(incex$occ), factor = 2),
     xlab = 'Occ. Exp.', ylab = 'Occ. Status',
     cex = .4, pch = 16, yaxt = 'n')
axis(2, 1:3, levels(incex$occ))  # add a y-axis
```

Boxplots (Split by a Categorical Variable)

```
boxplot(oexp ~ occ, data = incex, horizontal = T,
xlab = 'Occupational Experience', ylab = 'Occupational Status')
```

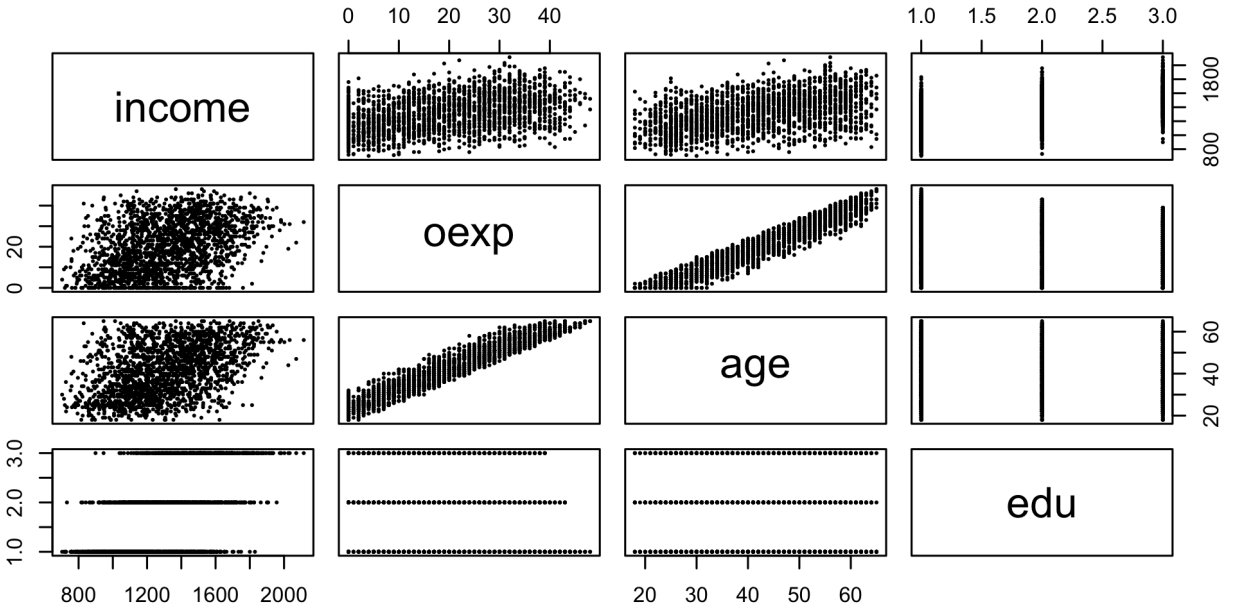


Jittered Scatterplot with Categorical Variables

```
plot(jitter(as.numeric(edu), factor = 2), jitter(as.numeric(occ), fac
     xlab = 'Edu. Level', ylab = 'Occ. Status', cex = .4, pch = 16,
     xaxt = 'n', yaxt = 'n')
axis(1, 1:3, levels(edu)); axis(2, 1:3, levels(occ))
```

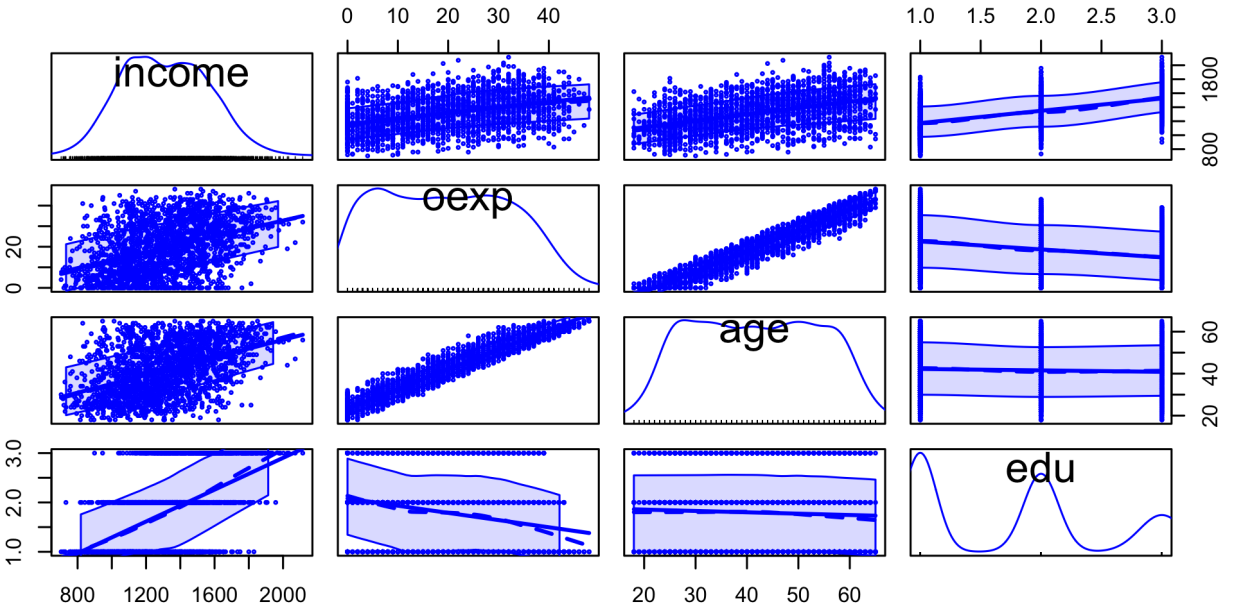
Scatterplot-Matrix

```
plot(incex[, c('income', 'oexp', 'age', 'edu')], cex = .2)
```



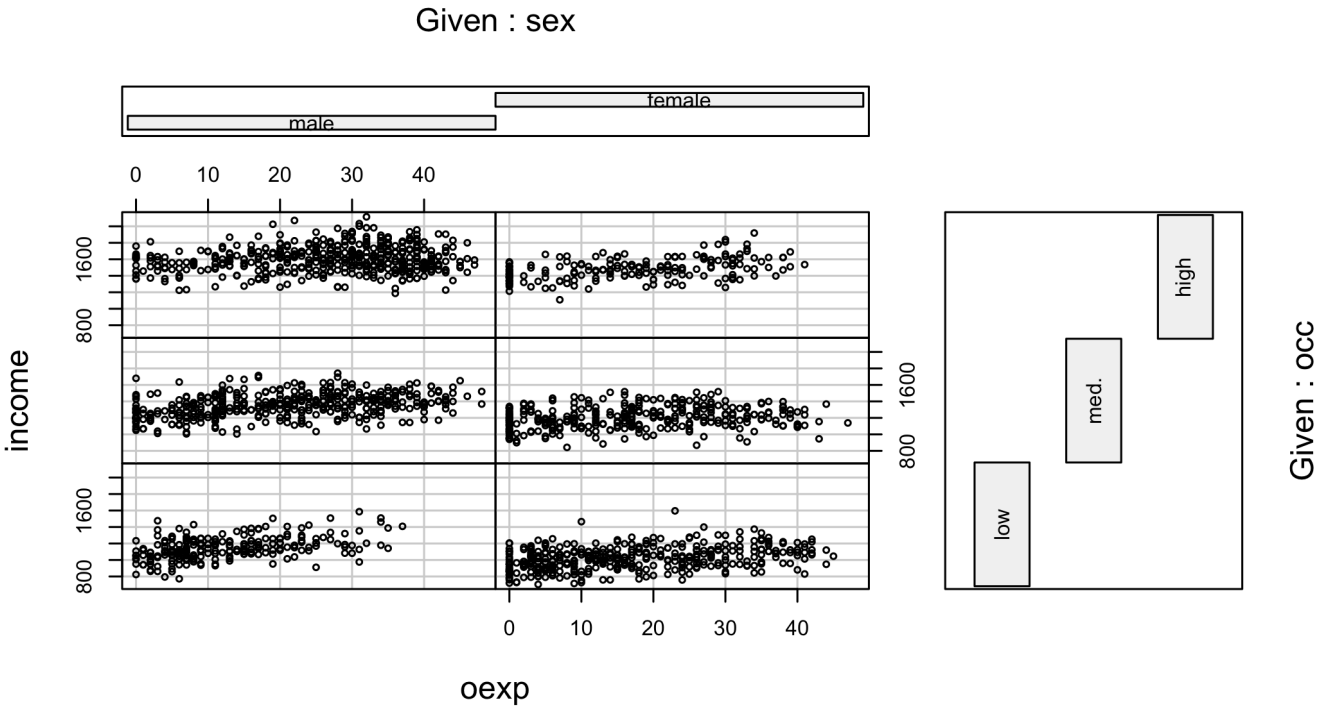
Scatterplot-Matrix

```
library(car)
scatterplotMatrix(incex[, c('income', 'oexp', 'age',
'edu')], cex = .3) # from the “car”-package
```



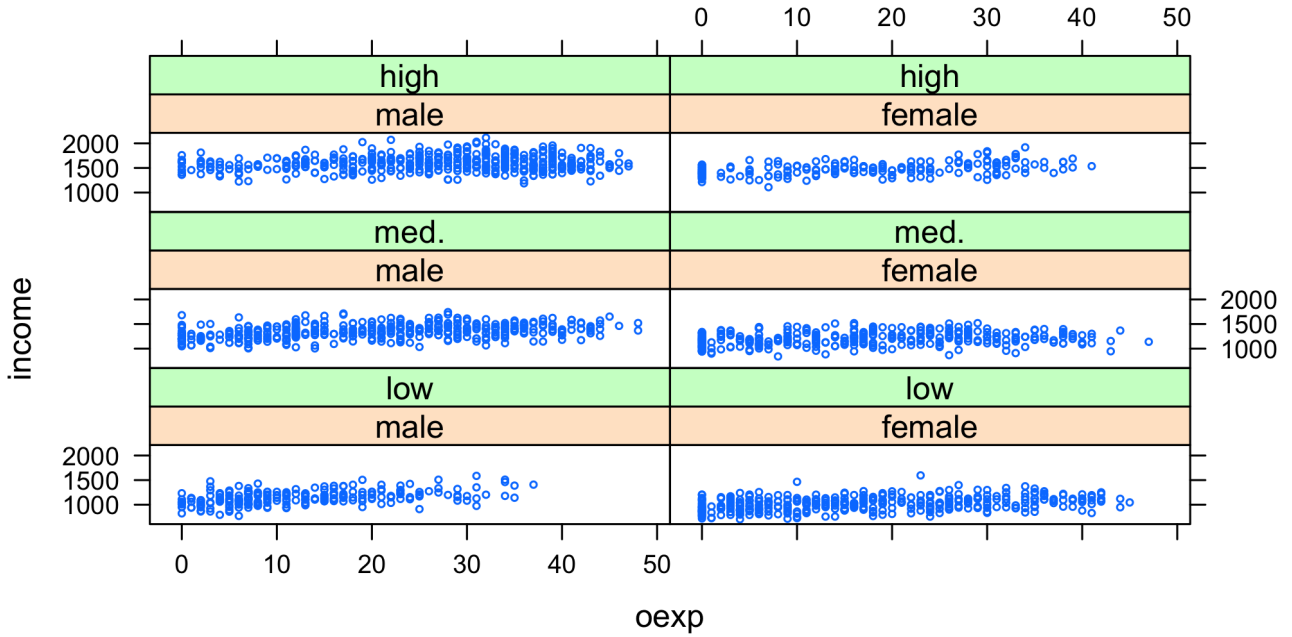
Conditioning Plots (coplot())

```
coplot(income ~ oexp | sex * occ, data = incex, cex = .6)
```



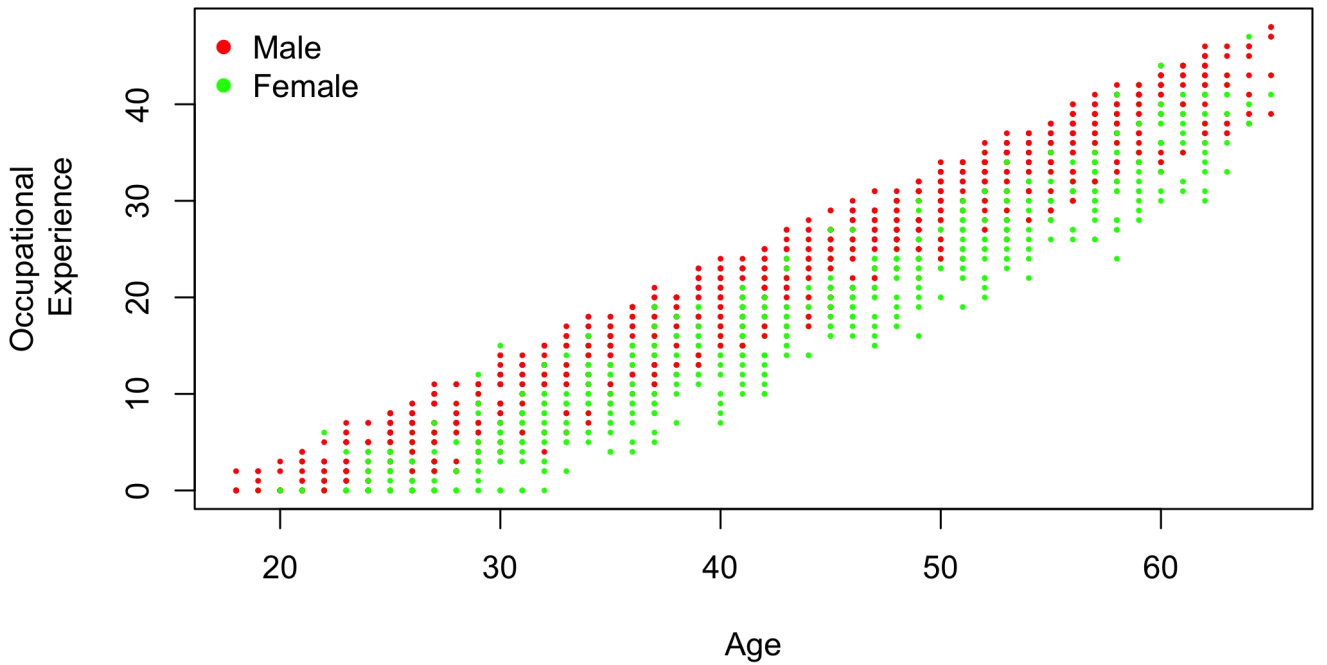
Conditioning Plots (xyplot() from the "lattice" package)

```
library(lattice)
xyplot(income ~ oexp | sex * occ, data = incex, cex = .4)
```



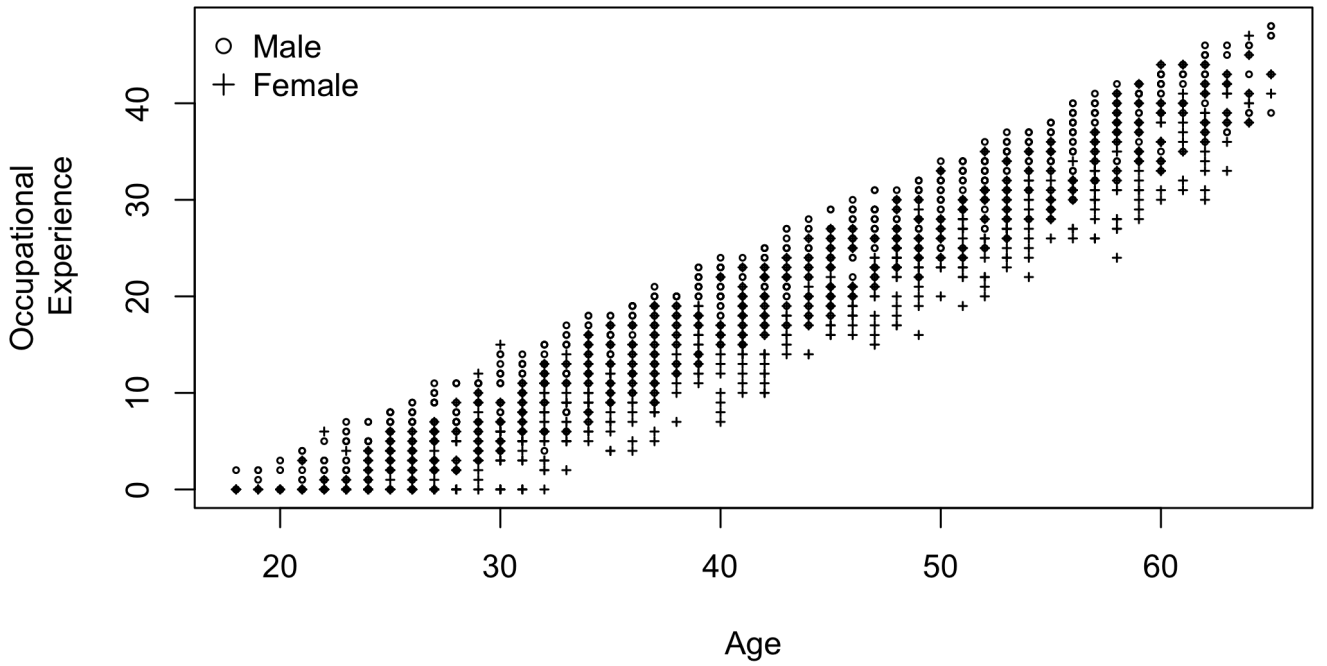
Scatterplot with a color separator

```
par(mar = c(4, 5, 1, 1))
plot(incex$age, incex$oexp, xlab = 'Age', ylab = 'Occupational
     Experience', cex = .4, pch = 16, col=c("red", "green")[as.numer
legend('topleft', c('Male', 'Female'), bty = 'n',
       col = c('red', 'green'), pch = c(16, 16))
```



Scatterplot with a shape separator

```
par(mar = c(4, 5, 1, 1))
plot(incex$age, incex$oexp, xlab = 'Age', ylab = 'Occupational
     Experience', cex = .4, pch = c(1, 3)[as.numeric(sex)])
legend('topleft', c('Male', 'Female'), bty = 'n', pch = c(1, 3))
```



- for more info on `pch` values, use the help `?pch`.

3-dim Scatterplot

```
# from "lattice" package
cloud(income ~ oexp * sex, data = incex, col = as.numeric(sex)+1)
```