Assessing Normality

STAT3022 Applied Linear Models Lecture 20

2020/02/18

THE UNIVERSITY OF SYDNEY

Today

1. Pearson's χ^2 test
2. Goodness of fit tests based on the empirical distribution function
3. Kolmogorov-Smirnov test, Cramer-von Mises test, Anderson-Darling test
4. R-package nortest
5. Shapiro-Wilk test, Shapiro-Francia test
6. Monte-Carlo p-values

Assessing normality

- So far we only ever assessed the normality assumption by inspection of Q-Q plots.
- When is it clear that the normality assumption is wrong?
- Can you think of any normality tests?

Data and testing problem

Given a data set $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ we want to test if there is evidence against H_0 that the data is a sample from a $N(\mu, \sigma^2)$ population.

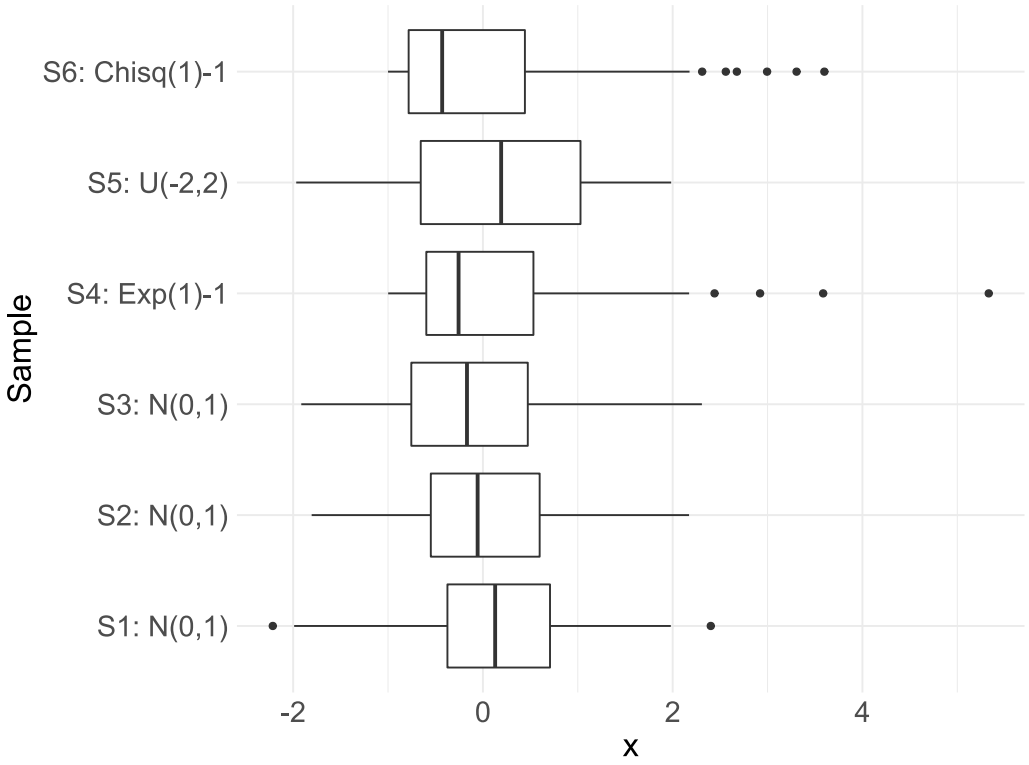
Example - Six pseudo-random samples of size $n = 64$

```
set.seed(1)
n <- 64
S1 <- rnorm(n, 0, 1)
S2 <- rnorm(n, 0, 1)
S3 <- rnorm(n, 0, 1)
S4 <- rexp(n) - 1
S5 <- 4 * runif(n, 0, 1) - 2
S6 <- rchisq(n, 1) - 1
dat <- data.frame(x=c(S1, S2, S3, S4, S5, S6),
                  Sample=rep(c("S1: N(0,1)",
                               "S2: N(0,1)",
                               "S3: N(0,1)",
                               "S4: Exp(1)-1",
                               "S5: U(-2,2)",
                               "S6: Chisq(1)-1"),
                             each=n))
```

```
head(dat,3) #simulated data
```

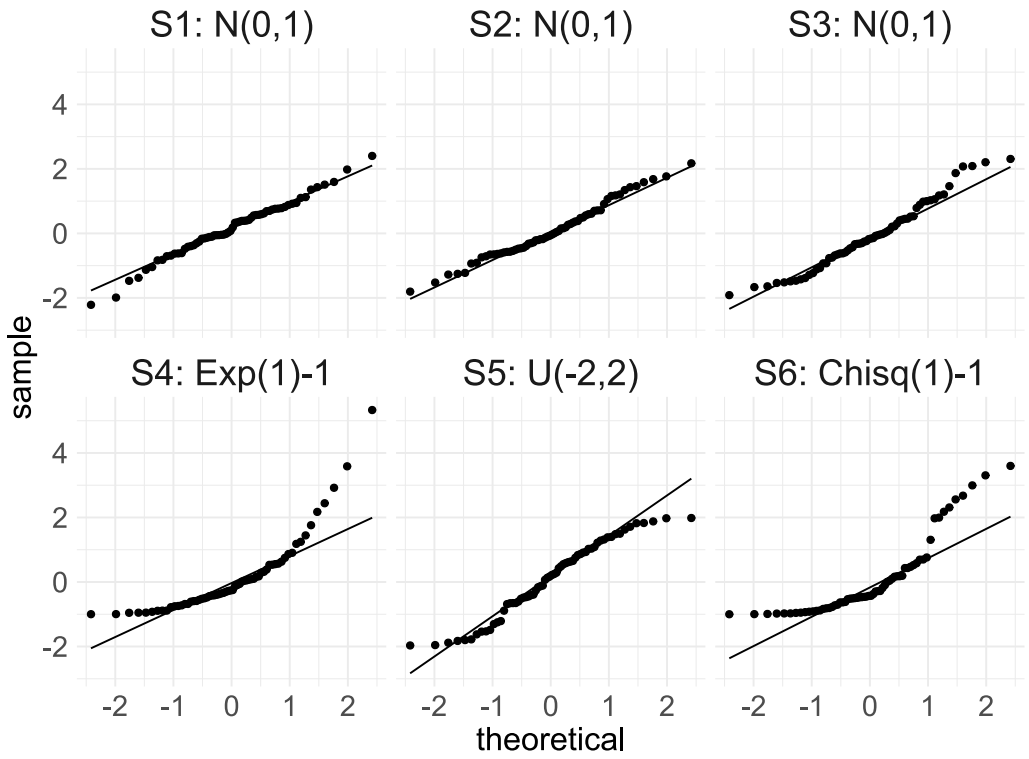
```
           x     Sample
1 -0.6264538 S1: N(0,1)
2  0.1836433 S1: N(0,1)
3 -0.8356286 S1: N(0,1)
```

Six pseudo-random samples: Boxplots



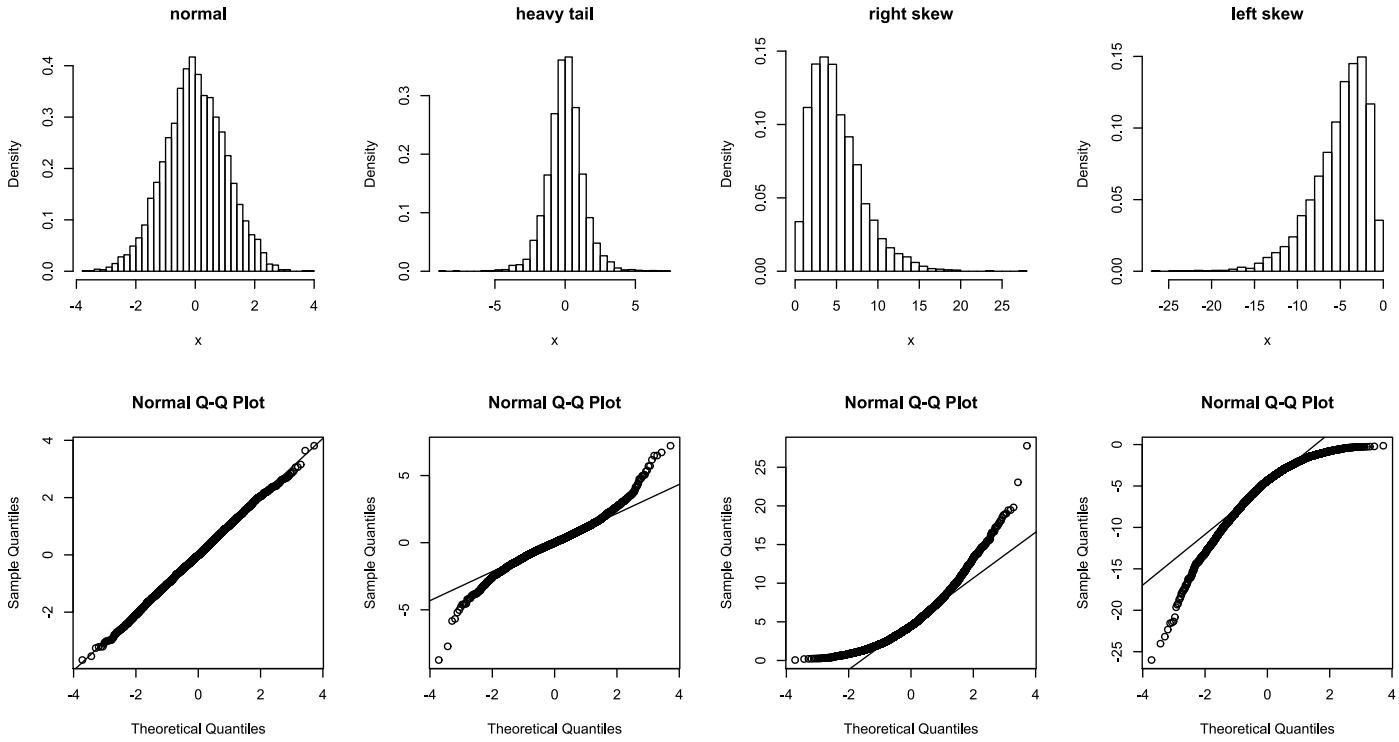
- You would look for symmetry and lack of outliers here.

Six pseudo-random samples: Q-Q-plots

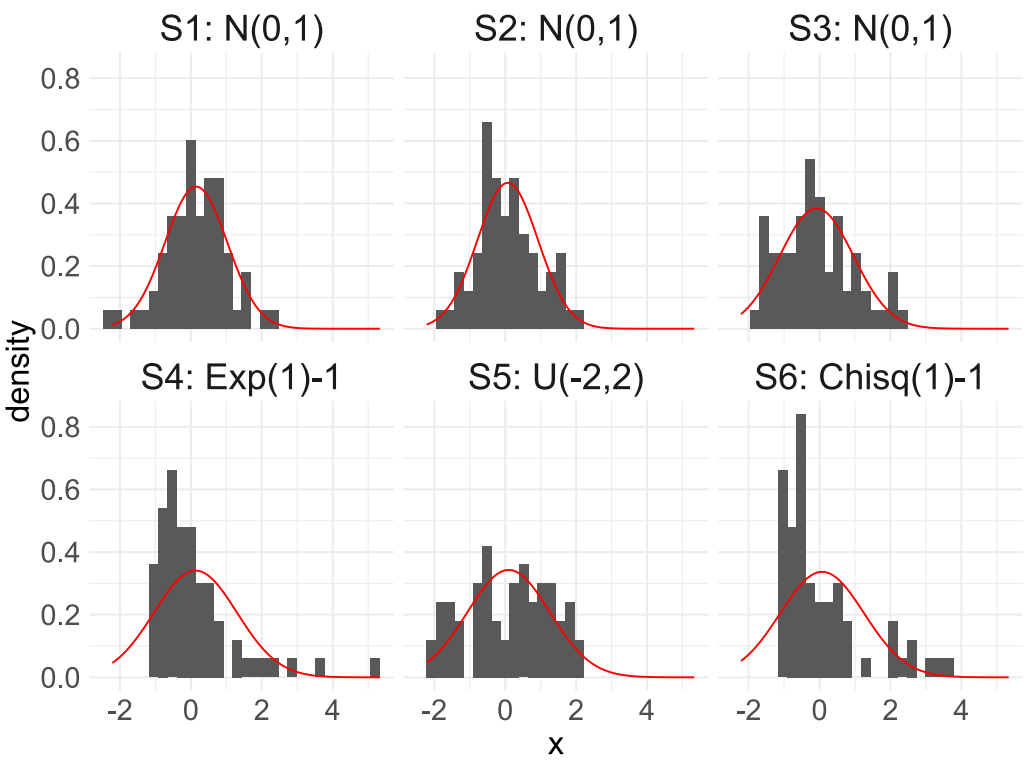


- You would want roughly a straight line here.
- S5 shows that the U(-2,2) is light-tailed.

How to interpret QQ plot



Six pseudo-random samples: Histogram



- Red lines are the density from $N(\bar{x}, s^2)$.
- You want the histogram to be simlar to the red normal curve.
- How to judge if it is not? You may want some numerical measures.

Pearson's χ^2 test

- First calculate the sample mean, $\bar{x}$, and the sample variance, s^2.
- Form a grouped frequency table summary of the data with say at most

$$g = \begin{cases} \lfloor n/5 \rfloor & n \leq 50 \\ \lceil 2 \times n^{2/5} \rceil & n \geq 50 \end{cases}$$

  categories. When $n = 64$, at most $\lceil 2 \times 64^{2/5} \rceil = 11$ categories.
- To check the normal claim, work out the expected frequencies (E_i) for each category by fitting $N(\bar{x}, s^2)$.
- Calculate the χ^2 test statistic with 2 estimates (lose 2df),

$$X^2 = \sum_{i=1}^{g} \frac{O_i^2}{E_i} - n \sim \chi^2_{g-2-1}.$$

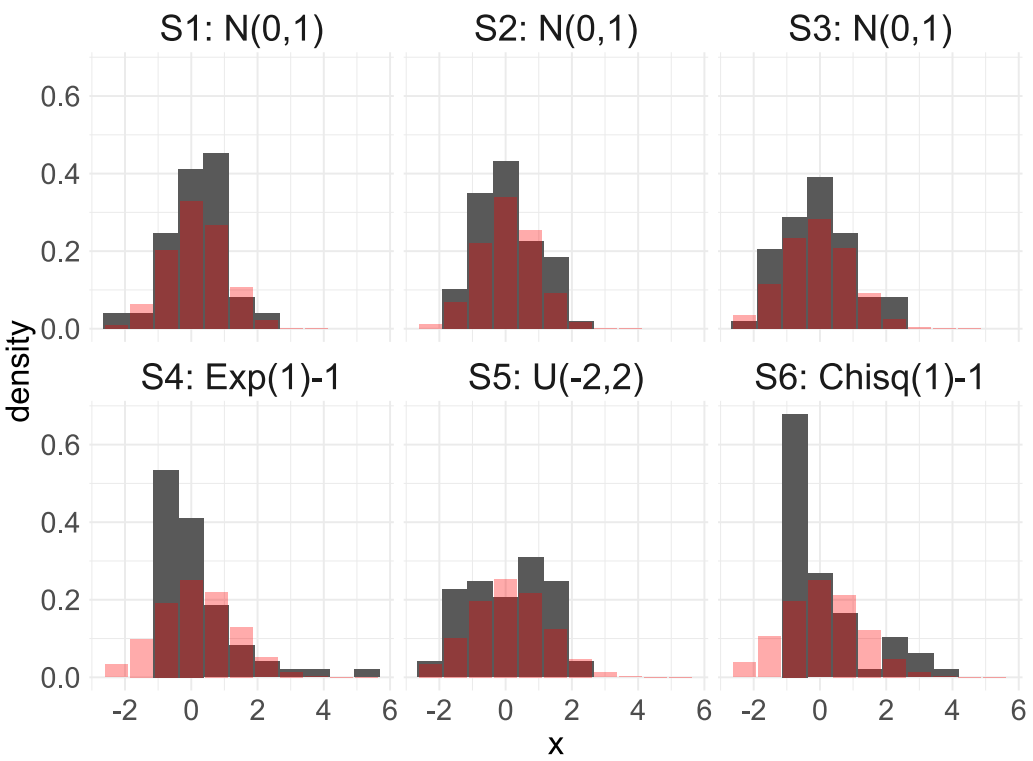
- In R with `pearson.test` in `library(nortest)` or very tediously by "hand".

```
data.frame(Sample=
    c("S1", "S2", "S3", "S4", "S5", "S6"),
    "P-value"=scales::pvalue(c(
     nortest::pearson.test(S1)$p.value,
     nortest::pearson.test(S2)$p.value,
     nortest::pearson.test(S3)$p.value,
     nortest::pearson.test(S4)$p.value,
     nortest::pearson.test(S5)$p.value,
    nortest::pearson.test(S6)$p.value)))
```

```
  Sample P.value
1     S1   0.081
2     S2   0.484
3     S3   0.520
4     S4  <0.001
5     S5   0.484
6     S6  <0.001
```

S5 is not significant!

Pearson's χ^2 test



- The black bars correspond to O_i.
- The red bars correspond to E_i under $N(\bar{x}, s^2)$.
- Pearson's χ^2 test is a special case of a normality test based on the empirical distribution (EDF) $\mathbb{F}_n$.
- Note that for $k = 1, \ldots, g$

$$E_k = n[F(u_k) - F(l_k)]$$

$$O_k = n[\mathbb{F}_n(u_k) - \mathbb{F}_n(l_k)],$$

  where F denotes the distribution under H_0.

Empirical distribution function tests

- Recall, that the EDF of a sample is a step function defined as

$$\mathbb{F}_n(x) = \begin{cases} 0 & x < x_{(1)} \\ i/n & x_{(i)} \le x < x_{(i+1)} \\ 1 & x_{(n)} \le x \end{cases} \qquad i = 1, \ldots, n-1$$

or

$$\mathbb{F}_n(x) = \begin{cases} 0 & x < x_{(1)} \\ (i-1)/n & x_{(i)} \le x < x_{(i+1)} \\ 1 - 1/n & x_{(n)} \ge x \end{cases} \qquad i = 1, \ldots, n$$

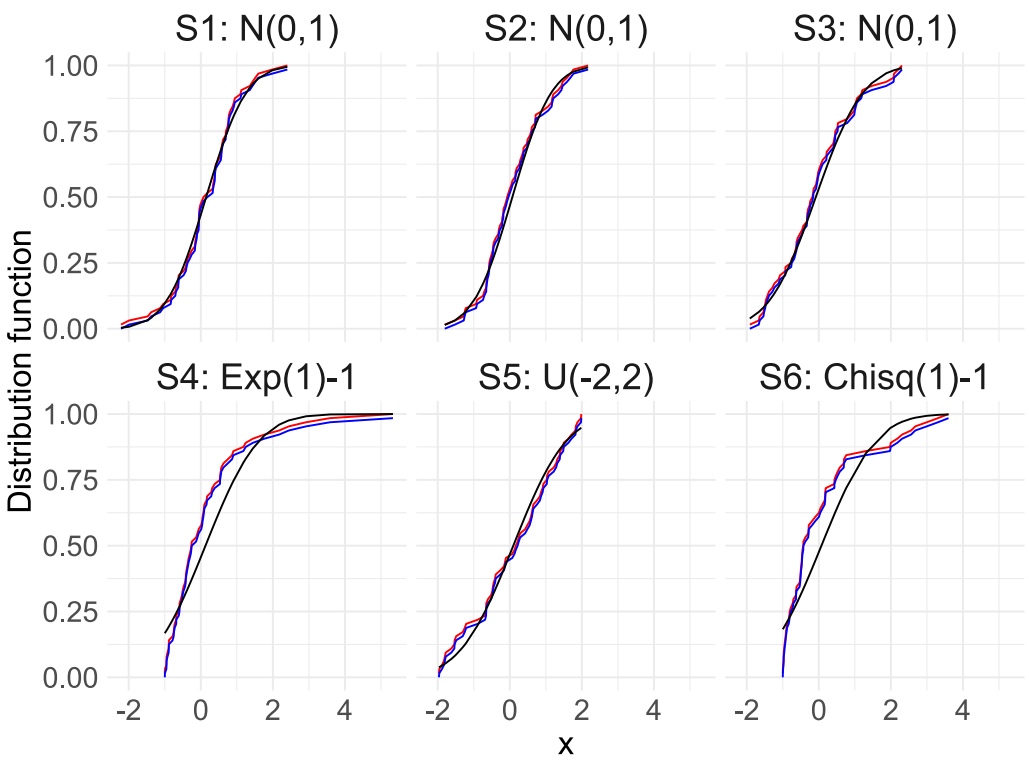
- EDF tests are based on "differences" between the EDF and the distribution function assumed under the null hypothesis.

- In the standard normal case $F(x) = \Phi(x)$.
- For $F = N(\mu, \sigma^2)$ we have to know or to estimate the parameters.
- For assessing normality of errors etc we have to at least estimate the error variance. In general we compare

$$p_{(i)} = \Phi\left(\frac{x_{(i)} - \hat{\mu}}{\hat{\sigma}}\right)$$

with values of $\mathbb{F}_n$, i.e. $(1/n, 2/n, \ldots, 1)$ or $(0, 1/n, \ldots, (n-1)/n)$.

Empirical distribution function




```
dat2 <- dat %>%
  group_by(Sample) %>%
  arrange(Sample, x) %>%
  mutate(
   pest=pnorm(x,mean(x),sd(x)),
   fest=(1:n)/n,
   fest2=(0:(n-1))/n)
ggplot(dat2) +
  geom_line(aes(x, fest),
            color="red") +
  geom_line(aes(x, fest2),
            color="blue") +
  facet_wrap(~Sample) +
  geom_line(aes(x, pest)) +
labs(y="Distribution function")
```

The Kolmogorov-Smirnov (KS) test

- The KS test statistic D is the maximum absolute differences between the EDF $\mathbb{F}_n$ and tested distribution function F with observed p_i as

$$D^+ = \max_{1\le i\le n}\{|i/n - p_{(i)}|\} \text{ and } D^- = \max_{1\le i\le n}\{|p_{(i)} - (i-1)/n|\}$$

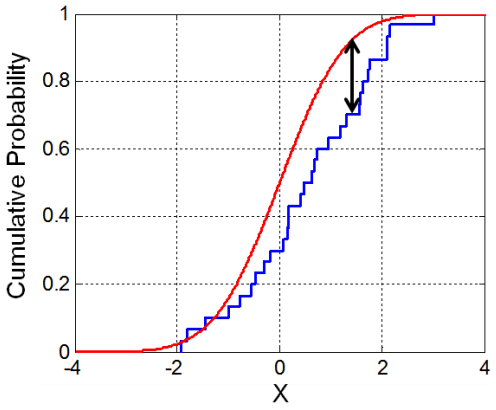
$$D = \max\{D^+, D^-\} \text{ and } \sqrt{n}D \xrightarrow{n\nearrow} K$$

- The Kolmogorov distribution K is independent of the distribution F being tested.
- In R `lillie.test(x)` in `library(nortest)` returns the same test statistic as `ks.test(x,"pnorm", mean=mean(x), sd=sd(x))` but the p-values differ.
- Reason: The p-value in `ks.test` is calculated under the assumption that mean and variance of the normal are known, thus its p-value is wrong when those parameters are estimated.
- Again, S5 is not significant.

```
data.frame(Sample=
 c("S1", "S2", "S3", "S4", "S5", "S6"),
 "P-value (Wrong)"=scales::pvalue(c(
 ks.test(S1, "pnorm", mean(S1),sd(S1))$p,
 ks.test(S2, "pnorm", mean(S2),sd(S2))$p,
 ks.test(S3, "pnorm", mean(S3),sd(S3))$p,
 ks.test(S4, "pnorm", mean(S4),sd(S4))$p,
 ks.test(S5, "pnorm", mean(S5),sd(S5))$p,
 ks.test(S6, "pnorm", mean(S6),sd(S6))$p)),
 "P-value"=scales::pvalue(c(
 nortest::lillie.test(S1)$p.value,
 nortest::lillie.test(S2)$p.value,
 nortest::lillie.test(S3)$p.value,
 nortest::lillie.test(S4)$p.value,
 nortest::lillie.test(S5)$p.value,
 nortest::lillie.test(S6)$p.value)))
```

```
  Sample P.value..Wrong. P.value
1     S1           0.928   0.705
2     S2           0.914   0.667
3     S3           0.882   0.589
4     S4           0.039  <0.001
5     S5           0.851   0.524
6     S6           0.013  <0.001
```


The KS test statistic D




```
Udif=abs(dat2$fest-dat2$pest); Dp=tapply(Udif,dat2$Sample,max)
Ldif=abs(dat2$pest-dat2$fest2); Dm=tapply(Ldif,dat2$Sample,max)
DD=cbind(Dp,Dm); D=apply(DD,1,max); cbind(DD,D)
```

```
                        Dp         Dm          D
S1: N(0,1)      0.05791344 0.06572799 0.06572799
S2: N(0,1)      0.06745749 0.06122743 0.06745749
S3: N(0,1)      0.07090000 0.05527500 0.07090000
S4: Exp(1)-1    0.17241070 0.16750493 0.17241070
S5: U(-2,2)     0.07280347 0.07381762 0.07381762
S6: Chisq(1)-1 0.19474035 0.18248152 0.19474035
```

```
print(head(dat3),digits=4) #fest=(1:n)/n,  fest2=(0:(n-1))/n)
```

```
        x     Sample     pest    fest   fest2     Udif     Ldif
1 -2.215 S1: N(0,1) 0.003579 0.01562 0.00000 0.012046 0.003579
2 -1.989 S1: N(0,1) 0.007488 0.03125 0.01562 0.023762 0.008137
3 -1.471 S1: N(0,1) 0.032670 0.04688 0.03125 0.014205 0.001420
4 -1.377 S1: N(0,1) 0.041254 0.06250 0.04688 0.021246 0.005621
5 -1.129 S1: N(0,1) 0.072904 0.07812 0.06250 0.005221 0.010404
6 -1.044 S1: N(0,1) 0.087307 0.09375 0.07812 0.006443 0.009182
```

Cramer-von Mises test

- In general the KS-test is a *conservative* test, ie it uses more information than needed to give the largest absolute difference between $\mathbb{F}_n$ and F.
- A class of EDF goodness of fit tests was proposed by Anderson and Darling and is defined by

$$n \int_{-\infty}^{\infty} [\mathbb{F}_n(x) - F(x)]^2 \psi(F(x))\, dF(x),$$

where $\psi(F(x))$ is a weighting function.

- Cramer and von Mises showed that choosing $\psi(F(x)) = 1$ and some calculus yields the *recommended* Cramer-von Mises test:

$$W = \frac{1}{12n} + \sum_{i=1}^{n} \left(p_{(i)} - \frac{2i-1}{2n} \right)^2.$$

- In R with `cvm.test(x)` in `library(nortest)`.

```
i=rep(1:n,6); pi=dat2$pest; w=(pi-(2*i-1)/(2*n))^2
sa=dat2$Sample; W=tapply(w,sa,sum)+1/(12*n)
data.frame(Sample=
 c("S1", "S2", "S3", "S4", "S5", "S6"),
 "P-value"=scales::pvalue(c(
   nortest::cvm.test(S1)$p.value,
   nortest::cvm.test(S2)$p.value,
   nortest::cvm.test(S3)$p.value,
   nortest::cvm.test(S4)$p.value,
   nortest::cvm.test(S5)$p.value,
   nortest::cvm.test(S6)$p.value)),"W"=W)
```

```
                Sample P.value          W
 S1: N(0,1)         S1   0.564 0.04616663
 S2: N(0,1)         S2   0.230 0.07598470
 S3: N(0,1)         S3   0.391 0.05845934
 S4: Exp(1)-1       S4  <0.001 0.62212939
 S5: U(-2,2)        S5   0.190 0.08225586
 S6: Chisq(1)-1     S6  <0.001 0.76314103
```

Again, S5 is not significant.

Anderson-Darling test

- Another recommended goodness of fit test is obtained by choosing a different weight function to the identity. Anderson and Darling proposed to use

$$\psi(F(x)) = \frac{1}{F(x) \times (1 - F(x))}$$

that is assigning much more weight to the upper and lower tail of the distribution under H_0.

- The Anderson-Darling test (`ad.test(x)` in `library(nortest)`) calculates

$$A = -n - \frac{1}{n} \sum_{i=1}^{n} [2i-1][\log(p_{(i)}) + \log(1 - p_{(n-i+1)})]$$

- Critical values for AD test depend on tested distribution and are tabulated for some dist.
- Again, S5 is not significant.

```
pr1=rev(pi[sa=="S1: N(0,1)"]); pr2=rev(pi[sa=="S2: N(0
pr3=rev(pi[sa=="S3: N(0,1)"]); pr4=rev(pi[sa=="S4: Exp
pr5=rev(pi[sa=="S5: U(-2,2)"]); pr6=rev(pi[sa=="S6: Ch
pr=c(pr1,pr2,pr3,pr4,pr5,pr6)
```

```
a=(2*i-1)*(log(pi)+log(1-pr)); A=-n-tapply(a,sa,sum)/n
data.frame(Sample=
 c("S1", "S2", "S3", "S4", "S5", "S6"),
 "P-value"=scales::pvalue(c(
   nortest::ad.test(S1)$p.value,
   nortest::ad.test(S2)$p.value,
   nortest::ad.test(S3)$p.value,
   nortest::ad.test(S4)$p.value,
   nortest::ad.test(S5)$p.value,
   nortest::ad.test(S6)$p.value)),"A"=A)
```

```
                Sample P.value         A
 S1: N(0,1)         S1   0.607 0.2881825
 S2: N(0,1)         S2   0.265 0.4519737
 S3: N(0,1)         S3   0.244 0.4665300
 S4: Exp(1)-1       S4  <0.001 3.6591840
 S5: U(-2,2)        S5   0.089 0.6430692
 S6: Chisq(1)-1     S6  <0.001 4.4832030
```

Shapiro-Wilk test

- Instead of using distances between $\mathbb{F}_n$ and F, Shapiro-Wilk test uses theoretical properties of the Q-Q-plot.
- If the sample came from a normal population, then the slope of the regression line of the ordered observations $x_{(i)}$'s against their expected values $m_i = E(X_{(i)})$'s of normal should be approximately 1 (the R^2 close to 1).
- The Shapiro-Wilk test statistic is

$$W = \frac{\left(\sum_{i=1}^{m} a_i x_{(i)}\right)^2}{S_{xx}}, \quad \text{where} \quad \boldsymbol{a} = \frac{\mathbf{V}^{-1}\boldsymbol{m}}{\sqrt{\boldsymbol{m}^\top \mathbf{V}^{-1}\mathbf{V}^{-1}\boldsymbol{m}}}$$

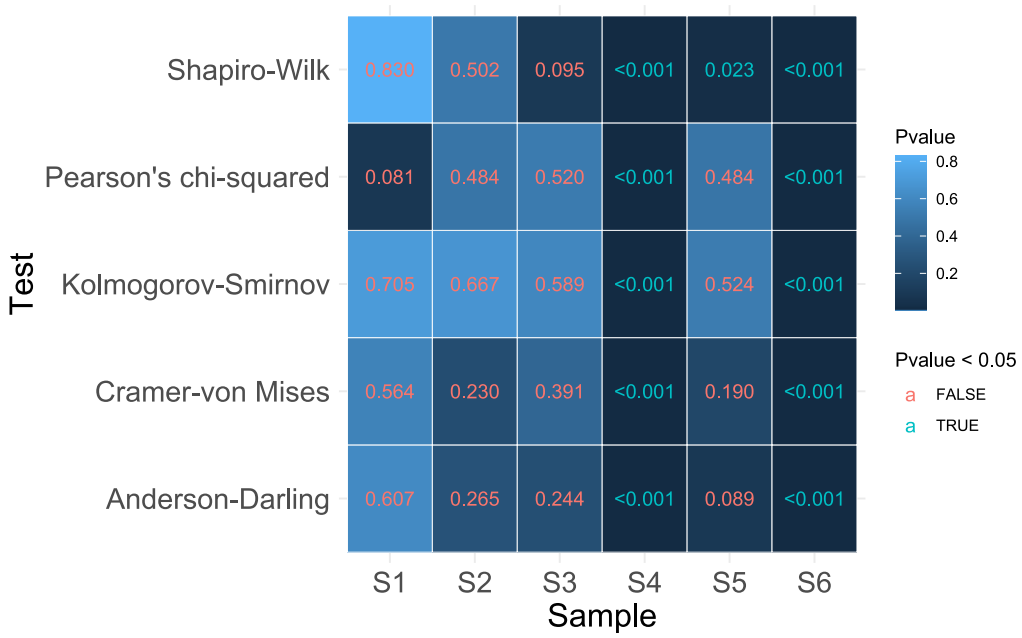
  and $\boldsymbol{m}$ and $\mathbf{V}$ being the vector of expected values and the covariance matrix of $(X_{(1)}, \ldots, X_{(n)})$ under the normality assumption.
- W is outputed but not evaluated since the covariance matrix $\mathbf{V}$ of $(X_{(1)}, \ldots, X_{(n)})$ is more complicated to calculated.

```
W=c(shapiro.test(S1)$statistic,shapiro.test
    shapiro.test(S4)$statistic,shapiro.test
data.frame(Sample=
 c("S1", "S2", "S3", "S4", "S5", "S6"),
 "P-value"=scales::pvalue(c(
   shapiro.test(S1)$p.value,
   shapiro.test(S2)$p.value,
   shapiro.test(S3)$p.value,
   shapiro.test(S4)$p.value,
   shapiro.test(S5)$p.value,
   shapiro.test(S6)$p.value)),"W"=W)
```

```
  Sample P.value         W
1     S1   0.830 0.9887678
2     S2   0.502 0.9825693
3     S3   0.095 0.9680163
4     S4  <0.001 0.7846641
5     S5   0.023 0.9560142
6     S6  <0.001 0.8002344
```

Only SW test gives significant result of S5 so far.

And the winner is ...



Shapiro-Wilk test!

How to get correct p-values?

- Think hard, study more (for a couple of years) and do the theory behind it.
- Or, use Monte-Carlo simulation to get resampling p-values, when the theory is too challenging!
- What is Monte-Carlo simulation?
- We show an example of a Monte-Carlo p-value using the last "famous" normality test known as the Shapiro-Francia test, which is simply the R^2 from the points in a Q-Q-plot.

Monte-Carlo and the R^2 of the Q-Q-plot

- Assume you have a data set $\boldsymbol{x}$ of size n with sample mean $\bar{x}$ and sample variance s^2 and from the points in the Q-Q-plot you obtain the corresponding $R^2 = r_*^2$.
- You want to find evidence against the sample coming from a normal population.
- To get a Monte-Carlo p-value, for say calculating the R^2 in the Q-Q-plot, calculate repeatedly, $b = 1, \ldots, B$ ($B = 1,000$ or larger), the following:
  - draw $\boldsymbol{x}_b$, a pseudo-normal sample from $N(\hat{\mu} = \bar{x}, \hat{\sigma}^2 = s^2)$;
  - calculate $r_b^2 = R^2$ of the Q-Q-plot of sample $\boldsymbol{x}_b$.
- The Monte-Carlo p-value is defined as

$$p_{\text{MC}} = \frac{1}{B} \sum_{i=1}^{B} \mathbb{I}\{r_b^2 < r_*^2\}.$$

Monte-Carlo p-value

```
set.seed(3)
B <- 1000
R <- matrix(0, B,6); rstar=rep(0,6)
out <- data.frame(Sample=paste0("S", 1:6),
                  Pvalue=NA)
m <- ppoints(n) # generates the expected values of
                # the order statistics used by the
                # qqnorm function to plot the
                # Q-Q-plot
for(i in 1:6) {
 asample <- paste0("S", i)
 xbar <- mean(get(asample))
 sdx <- sd(get(asample))
 rstar[i]<-cor(sort(get(asample)),qnorm(m,xbar,sdx))
 for (b in 1:B){
   xb  <- rnorm(n, xbar, sdx)
   R[b,i]<-cor(sort(xb), qnorm(m,xbar,sdx))
 }
 out[i, "Pvalue"] <- mean(R[,i]^2 < rstar[i]^2)
}
out=cbind(out,rstar^2)
```

```
out
```

```
  Sample Pvalue   rstar^2
1     S1  0.687 0.9871693
2     S2  0.544 0.9840466
3     S3  0.139 0.9722945
4     S4  0.000 0.7803068
5     S5  0.042 0.9632171
6     S6  0.000 0.8030071
```

So this test also gives significant result for S5 as SW test.

P-value is lower area from red line