Fundamentals of Statistical Modeling (VT21)

Andrea Discacciati
Karolinska Institutet
Stockholm, Sweden

Lab 2

Load the dataset and the `mlci` command

```
. version 14
. use https://raw.githubusercontent.com/anddis/fsm/master/data/lab2.dta, clear
. run https://raw.githubusercontent.com/anddis/fsm/master/do/mlci.do
```

Install the `qplot` command (you need to be connected to the Internet)

```
. net sj 16-3 gr42_7
--------------------------------------------------------------------------------
package gr42_7 from http://www.stata-journal.com/software/sj16-3
--------------------------------------------------------------------------------

TITLE
      SJ16-3 gr42_7. Update: Quantile plots
DESCRIPTION/AUTHOR(S)
      Update: Quantile plots
      by Nicholas J. Cox, Durham University,
           Department of Geography, Durham, UK
      Support:  n.j.cox@durham.ac.uk
      After installation, type help qplot
INSTALLATION FILES                                (type net install gr42_7)
      gr42_7/qplot.ado
      gr42_7/qplot.sthlp
--------------------------------------------------------------------------------

. net install gr42_7
checking gr42_7 consistency and verifying not already installed...
all files already exist and are up to date.
```

Install the `rcsgen` command (you need to be connected to the Internet)

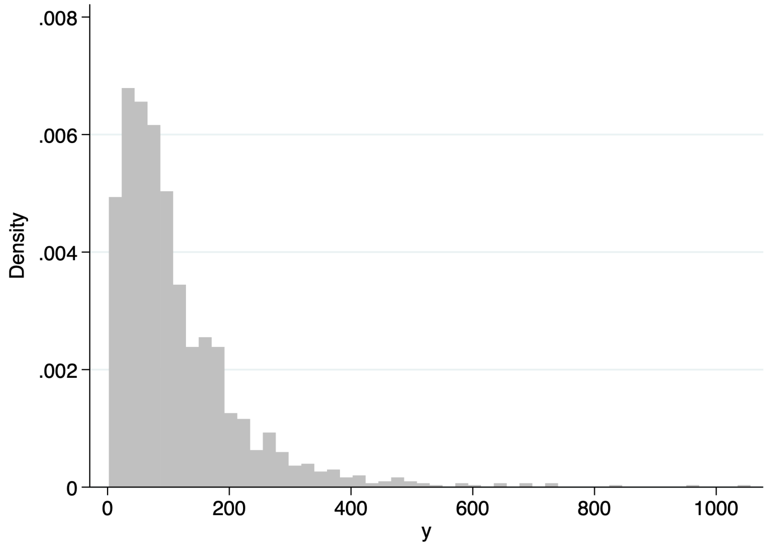
```
. cap net install http://fmwww.bc.edu/RePEc/bocode/r/rcsgen.pkg
```

Exercise 1

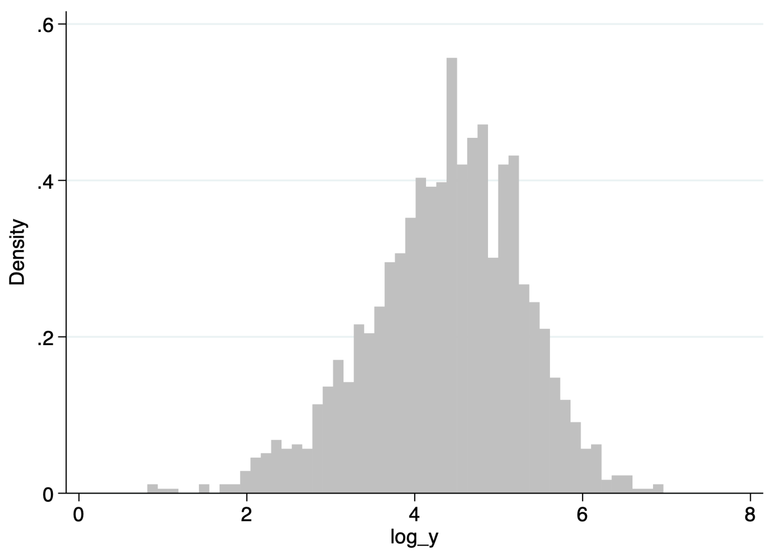
This dataset contains information on the blood concentration of a biomarker (y) in a random sample of 1432 subjects. Take a look at the histogram. What can we say about the distribution of this biomarker?

Plot also the histogram of $\log(y)$. How does the distribution of the biomarker after logarithmic transform look like?

```
. hist y, bin(50) name(p1, replace)
(bin=50, start=2.2592716, width=21.079531)
. graph export p1.png, replace
(file p1.png written in PNG format)
```




```
. gen log_y = log(y)

. hist log_y, bin(50) name(p2, replace)
(bin=50, start=.8150425, width=.12294848)

. graph export p2.png, replace
(file p2.png written in PNG format)
```



Exercise 2

We assume that $f(y)$ is gamma (see Lab 1). Estimate the parameters α and β using the `gammaden()` function. Fix the location parameter g (the third argument of the `gammaden()` function) to be equal to 0. Constrain α and β to be positive.

Note: the parameters α and β are not interpretable. We can reparametrise the gamma distribution so that one parameter is equal to its mean. This is described in the Extra material for Lab 2.

```
. local f = "gammaden(exp({theta1}), exp({theta2}), 0, y)"

. mlexp (log(`f´))

initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -96414.257
rescale:       log likelihood = -13891.173
rescale eq:    log likelihood = -13891.173
```

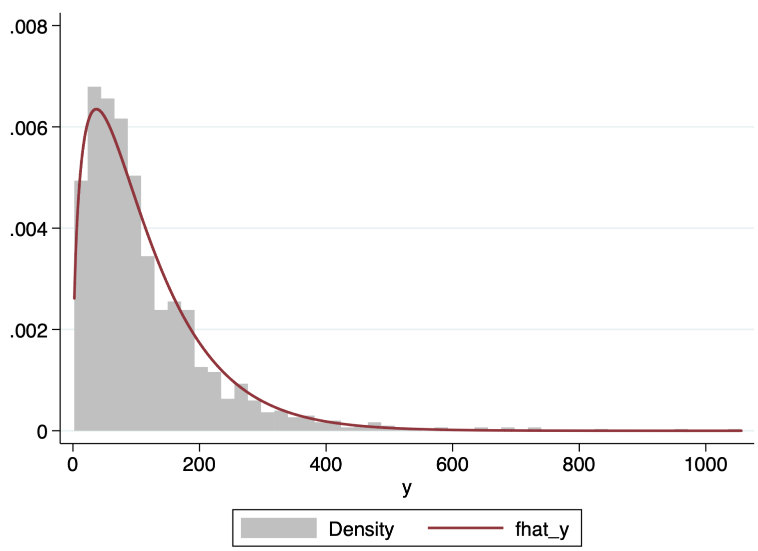
```
Iteration 0:   log likelihood = -13891.173
Iteration 1:   log likelihood = -8165.6417
Iteration 2:   log likelihood = -8160.8897
Iteration 3:   log likelihood = -8160.8781
Iteration 4:   log likelihood = -8160.8781
Maximum likelihood estimation
Log likelihood = -8160.8781                     Number of obs     =      1,432
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
/theta1	.3906123	.033984	11.49	0.000	.3240049	.4572196
/theta2	4.349872	.0403414	107.83	0.000	4.270804	4.42894

```
. mlci exp /theta1
 1.477885   95% CI: 1.382654, 1.579676
. mlci exp /theta2
 77.46854   95% CI: 71.57918, 83.84246
```

Plot the estimated density $\hat{f}(y)$ over the sample histogram

```
. gen fhat_y = gammaden(exp(_b[/theta1]), exp(_b[/theta2]), 0, y)
. tw (hist y, bin(50)) (line fhat_y y, sort), name(p3, replace) legend(rows(1))
. graph export p3.png, replace
(file p3.png written in PNG format)
```



Exercise 3

We assume that $f(y)$ is log-normal distributed. That is, we assume that the biomarker is standard normal distributed after we apply the transform

$$G(y) = (\log(y) - \mu)/\sigma$$

The derivative of $G(y)$ with respect to y is

$$G'(y) = g(y) = 1/(y\sigma).$$

Estimate the parameters μ and σ. Constrain σ to be positive.

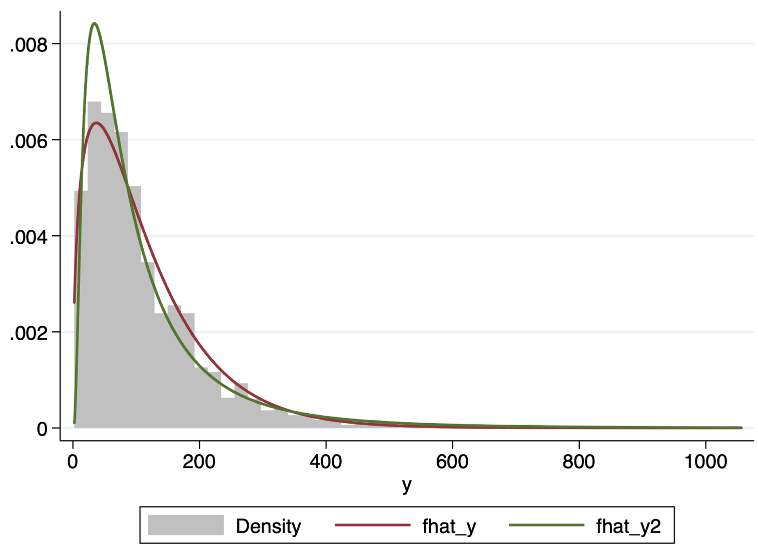
```
. local sigma = "exp({theta})"
. local G = "(log(y) - {mu}) / `sigma´"
. local g = "(1 / y / `sigma´)"
```

```
. local f = "normalden(`G´)*`g´"

. mlexp (log(`f´))

initial:       log likelihood = -21814.225
alternative:   log likelihood = -12440.421
rescale:       log likelihood = -10178.274
rescale eq:    log likelihood = -8264.4966
Iteration 0:   log likelihood = -8264.4966
Iteration 1:   log likelihood = -8198.7786
Iteration 2:   log likelihood = -8159.1222
Iteration 3:   log likelihood = -8158.8813
Iteration 4:   log likelihood = -8158.8812

Maximum likelihood estimation

Log likelihood = -8158.8812                    Number of obs     =      1,432

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         /mu |   4.365484   .0242269   180.19   0.000        4.318    4.412968
      /theta |  -.0868798   .0186859    -4.65   0.000    -.1235034   -.0502561
------------------------------------------------------------------------------

. mlci exp /theta
 .9167873   95% CI: .8838186, .9509858
```

Compare the likelihood with that from the gamma model

Plot the estimated density $\hat{f}(y)$ over the sample histogram

```
. gen fhat_y2 = normalden((log(y) - _b[/mu]) / exp(_b[/theta]))*(1 / y / exp(_b[/theta]))

. tw (hist y, bin(50)) (line fhat_y fhat_y2 y, sort), name(p4, replace) legend(rows(1))

. graph export p4.png, replace
(file p4.png written in PNG format)
```



Exercise 4

We make the transform $G(y)$ more flexible using polynomials. Consider the transform

$$G(y) = (\log(y) + \eta \log(y)^2 - \mu)/\sigma$$

The derivative of $G(y)$ with respect to y is

$$G'(y) = g(y) = (1 + 2\eta \log(y))/(\sigma y)$$

Estimate the parameters μ, σ, η. Constrain σ to be positive.

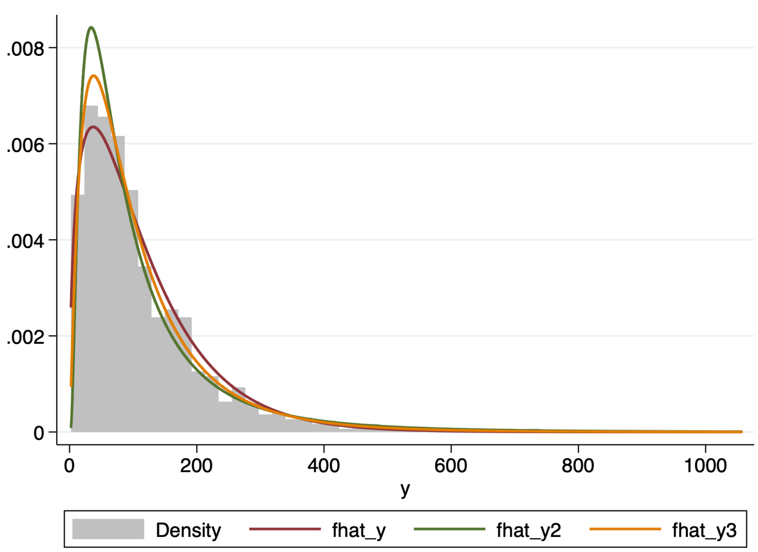
```
. local sigma = "exp({theta})"

. local G = "(log(y)+{eta}*log(y)^2 - {mu}) / `sigma´"

. local g = "(1 + {eta}*2*log(y)) / (`sigma´*y)"

. local f = "normalden(`G´)*`g´"

. mlexp (log(`f´))

initial:       log likelihood = -21814.225
alternative:   log likelihood = -62186.361
rescale:       log likelihood = -21814.225
rescale eq:    log likelihood = -21814.225
Iteration 0:   log likelihood = -21814.225  (not concave)
Iteration 1:   log likelihood = -10891.752  (not concave)
Iteration 2:   log likelihood = -8663.3365
Iteration 3:   log likelihood = -8355.3282
Iteration 4:   log likelihood = -8210.2164
Iteration 5:   log likelihood = -8167.6715
Iteration 6:   log likelihood = -8151.9226
Iteration 7:   log likelihood = -8142.8803
Iteration 8:   log likelihood = -8139.5952
Iteration 9:   log likelihood = -8138.8787
Iteration 10:  log likelihood = -8138.5163
Iteration 11:  log likelihood = -8138.5033
Iteration 12:  log likelihood = -8138.5022
Iteration 13:  log likelihood = -8138.5022

Maximum likelihood estimation

Log likelihood = -8138.5022                     Number of obs     =      1,432
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
/eta	.1804595	.0689773	2.62	0.009	.0452665	.3156524
/mu	7.956257	1.37386	5.79	0.000	5.263542	10.64897
/theta	.8361374	.2318207	3.61	0.000	.3817772	1.290498

```
. mlci exp /theta
 2.307437   95% CI: 1.464886, 3.634595
```

Plot the estimated density $\hat{f}(y)$ over the sample histogram

```
. gen fhat_y3 = normalden((log(y)+_b[/eta]*log(y)^2 - _b[/mu])/exp(_b[/theta])) * ///
>  (1+_b[/eta]*2*log(y)) / (exp(_b[/theta]) * y)

. tw (hist y, bin(50)) (line fhat_y fhat_y2 fhat_y3 y, sort), name(p5, replace) legend(rows(1))

. graph export p5.png, replace
(file p5.png written in PNG format)
```



Exercise 5

Instead of a quadratic term, we add two restricted cubic splines transforms of $\log(y)$: $V_2(\log(y))$ and $V_3(\log(y))$. We consider the transform

$$G(y) = (\log(y) + \eta_1 V_2(\log(y)) + \eta_2 V_3(\log(y)) - \mu)/\sigma$$

The derivative of $G(y)$ with respect to y is

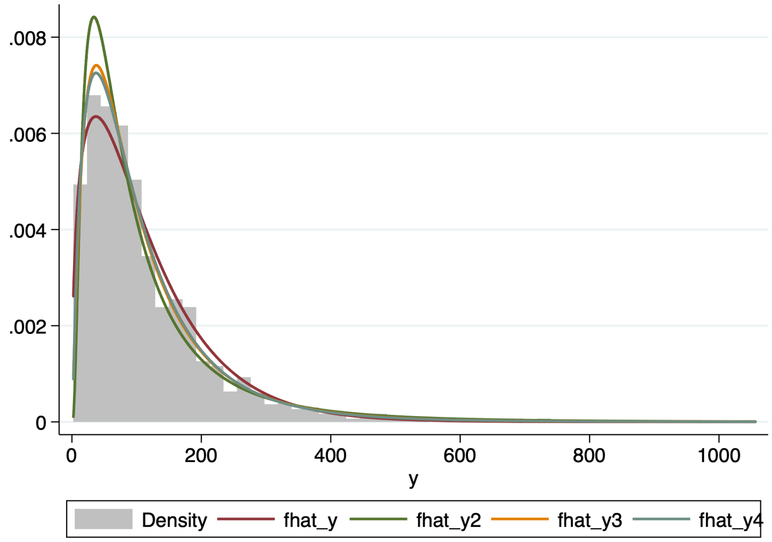
$$G'(y) = g(y) = (1 + \eta_1 v_2(\log(y)) + \eta_2 v_3(\log(y)))\,/(\sigma y)$$

Estimate the parameters $\mu, \sigma, \eta_1, \eta_2$. Constrain σ to be positive. Jointly test the 2 parameters η_1, η_2 to assess whether adding the 2 RCS transforms improves the fit of this model with respect to the "basic" log-normal model (see Exercise 3).

```
. rcsgen log_y, gen(V) dgen(v) df(3)
Variables V1 to V3 and v1 to v3 were created

. local sigma = "exp({theta})"

. local G = "(log(y)+{eta1}*V2+{eta2}*V3-{mu})/`sigma´"

. local g = "(1+{eta1}*v2+{eta2}*v3)/(`sigma´*y)"

. local f = "normalden(`G´)*`g´"

. mlexp (log(`f´))

initial:       log likelihood = -21814.225
final:         log likelihood = -21814.225
rescale:       log likelihood = -21814.225
Iteration 0:   log likelihood = -21814.225  (not concave)
Iteration 1:   log likelihood = -15749.959  (not concave)
Iteration 2:   log likelihood = -11575.393  (not concave)
Iteration 3:   log likelihood = -8930.8791  (not concave)
Iteration 4:   log likelihood =  -8353.802
Iteration 5:   log likelihood = -8226.7656
Iteration 6:   log likelihood = -8161.3477
Iteration 7:   log likelihood = -8141.3375
Iteration 8:   log likelihood =  -8137.886
Iteration 9:   log likelihood =  -8137.333
Iteration 10:  log likelihood = -8137.3305
Iteration 11:  log likelihood = -8137.3305

Maximum likelihood estimation

Log likelihood = -8137.3305                     Number of obs     =      1,432

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       /eta1 |  -.0080713   .0421601    -0.19   0.848    -.0907036     .074561
       /eta2 |  -.0259272   .0451895    -0.57   0.566     -.114497    .0626426
         /mu |   5.045942   .2441727    20.67   0.000     4.567372    5.524512
      /theta |   .2911407   .1009571     2.88   0.004     .0932684     .489013
------------------------------------------------------------------------------

. mlci exp /theta
 1.337953   95% CI: 1.097756, 1.630706

.
. test  [eta1]_cons [eta2]_cons

 ( 1)  [eta1]_cons = 0
 ( 2)  [eta2]_cons = 0

           chi2(  2) =   20.73
         Prob > chi2 =    0.0000
```

Plot the estimated density $\hat{f}(y)$ over the sample histogram

```
. gen fhat_y4 = normalden((log(y)+_b[/eta1]*V2+_b[/eta2]*V3 - _b[/mu])/exp(_b[/theta])) * ///
>  (1+_b[/eta1]*v2+_b[/eta2]*v3) / (exp(_b[/theta]) * y)

. tw (hist y, bin(50)) (line fhat_y fhat_y2 fhat_y3 fhat_y4 y, sort), name(p6, replace) legend(rows(1))

. graph export p6.png, replace
(file p6.png written in PNG format)
```



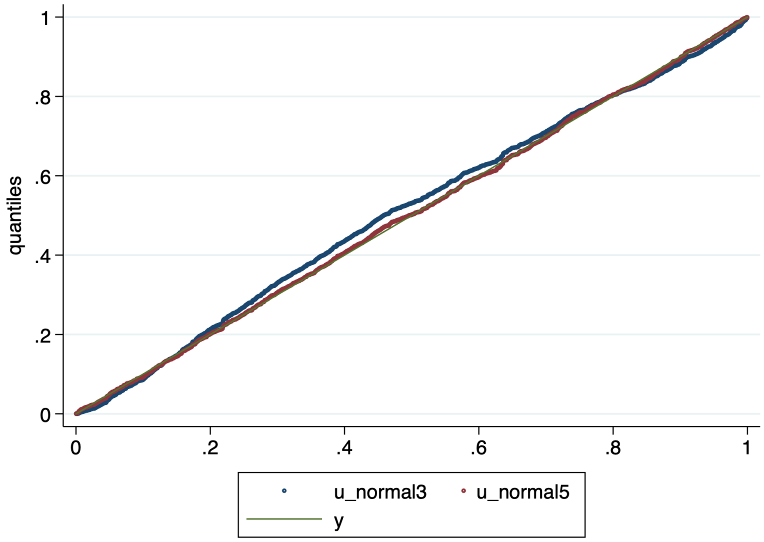
Exercise 6

Let's assess the goodness-of-fit of the log-normal model with RCS transforms (see Exercise 5) and of the log-normal model (see Excercise 3) using a quantile plot.

```
. gen u_normal5 = normal((log(y)+_b[/eta1]*V2+_b[/eta2]*V3 - _b[/mu])/exp(_b[/theta]))
.
. // Re-fit log-normal model (Exercise 3)
. local sigma = "exp({theta})"
. local G = "(log(y) - {mu}) / `sigma´"
. local g = "(1 / y / `sigma´)"
. local f = "normalden(`G´)*`g´"
. mlexp (log(`f´))
initial:       log likelihood = -21814.225
alternative:   log likelihood = -12440.421
rescale:       log likelihood = -10178.274
rescale eq:    log likelihood = -8264.4966
Iteration 0:   log likelihood = -8264.4966
Iteration 1:   log likelihood = -8198.7786
Iteration 2:   log likelihood = -8159.1222
Iteration 3:   log likelihood = -8158.8813
Iteration 4:   log likelihood = -8158.8812
Maximum likelihood estimation
Log likelihood = -8158.8812                     Number of obs     =      1,432
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
/mu	4.365484	.0242269	180.19	0.000	4.318	4.412968
/theta	-.0868798	.0186859	-4.65	0.000	-.1235034	-.0502561

```
. gen u_normal3 = normal((log(y) - _b[/mu])/exp(_b[/theta]))
.
. qplot u_normal3 u_normal5, addplot(function y = x, lw(medthin)) name(p7, replace) ///
>  msym(Oh Oh) msize(tiny tiny)
. graph export p7.png, replace
(file p7.png written in PNG format)
```



Extra: Exercise 7 (more on transforms of random variables)

We now assume that $f(y)$ is gamma-distributed after square root transform.

$$G(y) = \sqrt{y}$$

The derivative is

$$G'(y) = g(y) = 0.5/\sqrt{y}$$

Estimate the parameters α and β using the `gammaden()` function. Fix the location parameter g to be equal to 0. Constrain α and β to be positive. Compare the likelihood with that form the log-normal and gamma models

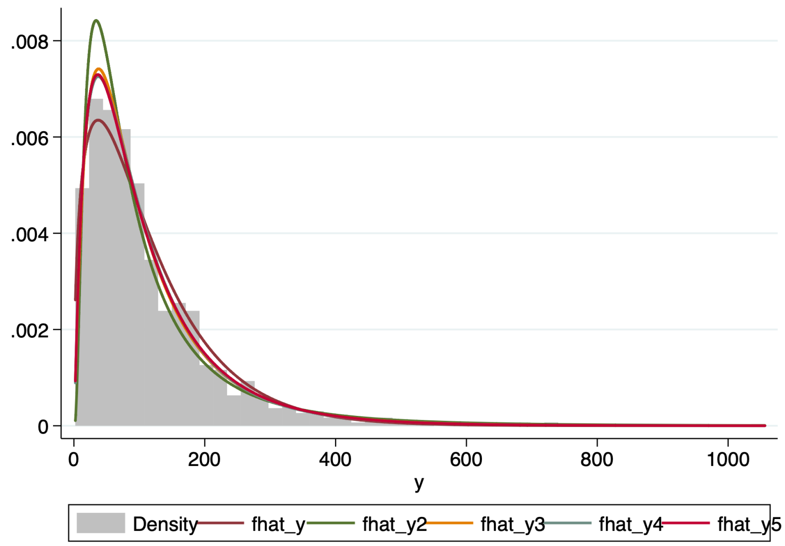
```
. local G = "sqrt(y)"
. local g = "(0.5 / sqrt(y))"
. local f = "gammaden(exp({theta1}),exp({theta2}),0,`G´)*`g´"
. mlexp (log(`f´))
initial:       log likelihood = -18140.526
alternative:   log likelihood = -11624.987
rescale:       log likelihood = -8442.1703
rescale eq:    log likelihood = -8442.1703
Iteration 0:   log likelihood = -8442.1703
Iteration 1:   log likelihood = -8185.3624
Iteration 2:   log likelihood = -8138.5993
Iteration 3:   log likelihood = -8138.2943
Iteration 4:   log likelihood = -8138.2942
Maximum likelihood estimation
Log likelihood = -8138.2942                      Number of obs     =      1,432
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
/theta1	1.652508	.0362406	45.60	0.000	1.581478	1.723538
/theta2	.6290655	.0380415	16.54	0.000	.5545055	.7036256

```
. mlci exp /theta1
 5.220056   95% CI: 4.862136, 5.604323
. mlci exp /theta2
 1.875857   95% CI:  1.74108, 2.021067
```

Plot the estimated density $\hat{f}(y)$ over the sample histogram. Visually compare the estimated density from the lognormal + splines model with the density from the gamma model after square root transform. What do you conclude?

```
. gen fhat_y5 = gammaden(exp(_b[/theta1]), exp(_b[/theta2]), 0, sqrt(y))*(.5 / sqrt(y))
. tw (hist y, bin(50)) (line fhat_y fhat_y2 fhat_y3 fhat_y4 fhat_y5 y, sort), name(p8, replace) legend(rows(1))
. graph export p8.png, replace
(file p8.png written in PNG format)
```



Extra: Exercise 8 (more on goodness of fit)

Let's go back to the normal distributed variable (Exercise 1, Lab 1).

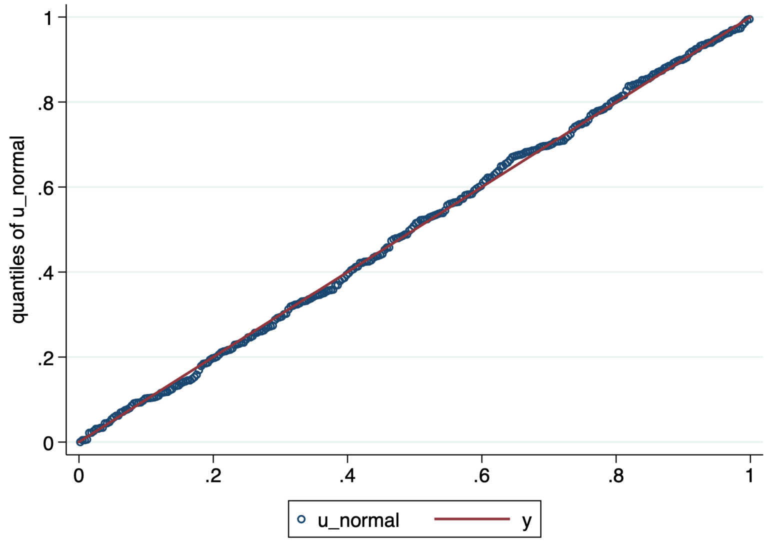
```
. use https://raw.githubusercontent.com/anddis/fsm/master/data/lab1.dta, clear
```

Assume that $f(y_n)$ is normal and estimate the parameters μ and σ. Generate the transform $u = \hat{F}(y_n)$. Draw the estimated quantile plot using the `qplot` command.

```
. local f = "normalden(y_n, {mu}, exp({theta}))"
. mlexp(ln(`f´))
initial:       log likelihood =     -<inf>  (could not be evaluated)
feasible:      log likelihood = -32398.765
rescale:       log likelihood = -1981.1218
rescale eq:    log likelihood = -1440.3171
Iteration 0:   log likelihood = -1440.3171  (not concave)
Iteration 1:   log likelihood = -1112.6119
Iteration 2:   log likelihood = -1085.7986
Iteration 3:   log likelihood = -1059.4172
Iteration 4:   log likelihood =  -1059.332
Iteration 5:   log likelihood = -1059.3319
Maximum likelihood estimation
Log likelihood = -1059.3319                     Number of obs     =        300
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
/mu	178.4911	.4772459	374.00	0.000	177.5557	179.4265
/theta	2.112168	.0408248	51.74	0.000	2.032153	2.192183

```
. mlci exp /theta
 8.266141   95% CI: 7.630494, 8.954739
.
. gen u_normal = normal((y_n-_b[/mu])/exp(_b[/theta]))
. qplot u_normal, addplot(function y = x) name(p1, replace)
. graph export p0.png, replace
(file p0.png written in PNG format)
```

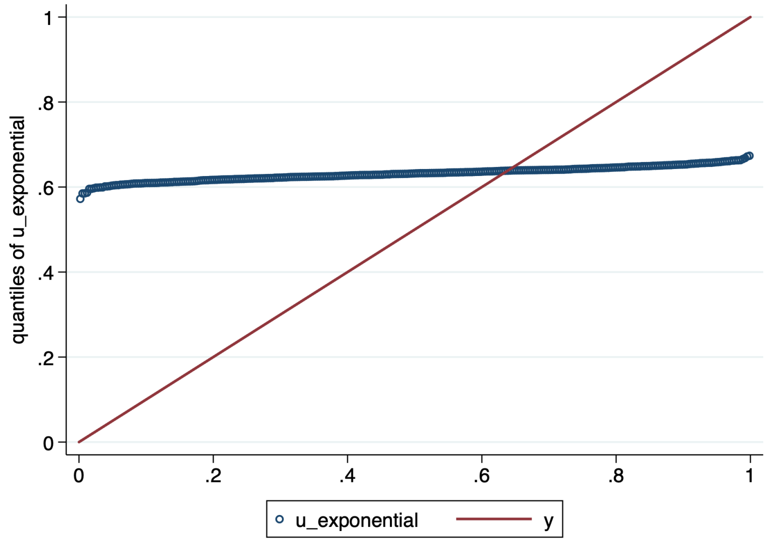


Assume now that $f(y_n)$ is exponential and estimate the parameter λ. Generate the transform $u = \hat{F}(y_n)$. Draw the estimated quantile plot using the `qplot` command.

```
. local f = "exp({theta})*exp(-y_n * exp({theta}))"

. mlexp(ln(`f´))

initial:       log likelihood = -53547.324
alternative:   log likelihood = -32628.094
rescale:       log likelihood = -2180.7534
Iteration 0:   log likelihood = -2180.7534
Iteration 1:   log likelihood = -1857.2088
Iteration 2:   log likelihood = -1855.3682
Iteration 3:   log likelihood = -1855.3616
Iteration 4:   log likelihood = -1855.3616

Maximum likelihood estimation

Log likelihood = -1855.3616                     Number of obs     =        300
```

	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
/theta	-5.184539	.057735	-89.80	0.000	-5.297697	-5.07138

```
. mlci exp /theta
 .0056025   95% CI: .0050031, .0062738

.
. gen u_exponential = 1-exp(-y_n * exp(_b[/theta]))

. qplot u_exponential, addplot(function y = x) name(p2, replace)

. graph export p00.png, replace
(file p00.png written in PNG format)
```



What can you conclude about the goodness of fit of the normal and exponential model?

Extra: Exercise 9 (binary variables)

Assume that y_{ber} follows a Bernoulli distribution. We want to estimate the probability of “success” ($y_{ber} = 1$). Estimate the probability η while constraining it to be bounded between 0 and 1. First, write down the likelihood by hand. Then, use the `binomialp()` function.

Are the results you obtain identical to those obtained from logistic regression?

```
. local eta = "invlogit({theta})"
. local f = "`eta´^y_ber * (1-`eta´)^(1-y_ber)"
. mlexp (ln(`f´))
initial:       log likelihood = -207.94415
alternative:   log likelihood =  -199.7231
rescale:       log likelihood =  -199.7231
Iteration 0:   log likelihood =  -199.7231
Iteration 1:   log likelihood = -199.70172
Iteration 2:   log likelihood = -199.70172
Maximum likelihood estimation
Log likelihood = -199.70172                     Number of obs     =        300

             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      /theta |   .4754237   .1187479     4.00   0.000     .2426821    .7081653
------------------------------------------------------------------------------
. mlci invlogit /theta
 .6166667   95% CI: .5603745, .6699956
.
. local eta = "invlogit({theta})"
. local f = "binomialp(1, y_ber, `eta´)"
. mlexp (ln(`f´))
initial:       log likelihood = -207.94415
alternative:   log likelihood =  -199.7231
rescale:       log likelihood =  -199.7231
Iteration 0:   log likelihood =  -199.7231
Iteration 1:   log likelihood = -199.70172
Iteration 2:   log likelihood = -199.70172
Maximum likelihood estimation
Log likelihood = -199.70172                     Number of obs     =        300

             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
```

```
      /theta |   .4754237   .1187479     4.00   0.000     .2426821    .7081653
------------------------------------------------------------------------------

. mlci invlogit /theta
 .6166667   95% CI: .5603745, .6699956

.
. logit y_ber

Iteration 0:   log likelihood = -199.70172
Iteration 1:   log likelihood = -199.70172

Logistic regression                               Number of obs     =        300
                                                  LR chi2(0)        =       0.00
                                                  Prob > chi2       =          .
Log likelihood = -199.70172                       Pseudo R2         =     0.0000

------------------------------------------------------------------------------
       y_ber |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       _cons |   .4754237   .1187479     4.00   0.000     .2426821    .7081653
------------------------------------------------------------------------------
```