```
function f = sample_from_dist(Data0, bins, seed)
% Generates binned data for the probability distribution function of Data0,
% creates DataNew by sampling from the pdf, and generates binned data for
% the pdf of DataNew.

%% Determining pdf and generating samples

% i) ----------------------------------------------------------------]
n = length(Data0);
[N0, edges0] = histcounts(Data0, bins);
%^% Generates bins, and the frequency of data within each
Data0_pdf = N0 / n; % Such that the total count is equivalent to P = 1

% ii) ---------------------------------------------------------------]
Data0_cdf = zeros(1, bins+1); % Initialises a 1x20 array
for i = 1:bins
    Data0_cdf(i+1) = sum(Data0_pdf(1:i));
    %^% Sums elements from 1 to i of Data0_pdf, such that they add up to 1.
    % Changed to (i+1), such that there is an element zero
end

rng(seed); % Sets random number generation to a specific seed
DataNew = zeros(1,n); % Initialising generated sample
for sample = 1:n
    u = rand; % Generate random number between 0 and 1
    for i = 2:bins+1 % Starting at i=2, since Data0_cdf(1) = 0
        if u <= Data0_cdf(i) % Determining which bin this sample is from
            x1 = edges0(i-1); x2 = edges0(i); % Segment of data this bin contains
            y1 = Data0_cdf(i-1); y2 = Data0_cdf(i); % Segment of cdf
            % y1 < u <= y2; to convert this into a data sample, we find the
            % x coordinate associated with u.
            DataNew(sample) = (u - y1) * ((x2 - x1) / (y2 - y1)) + x1;
            break
        end
    end
end

% iii) ---------------------------------------------------------------]
[N1, edges1] = histcounts(DataNew, bins);
DataNew_pdf = N1 / n; % such that the total count is equivalent to P = 1

%% Kullback-Leibler measure

% iv) ----------------------------------------------------------------]
DKL_step = @(p, q) p * log(p/q);
DKL_0 = 0;
for i = 1:bins
    step = DKL_step(Data0_pdf(i), DataNew_pdf(i));
    if isfinite(step) && ~isnan(step)
        DKL_0 = DKL_0 + step;
    else
```

```
        % if p or q are zero, output step but do not add it
        step
    end
end
DKL_New = 0;
for i = 1:bins
    step = DKL_step(DataNew_pdf(i), Data0_pdf(i));
    if isfinite(step) && ~isnan(step)
        DKL_New = DKL_New + step;
    else
        % if p or q are zero, output step but do not add it
        step
    end
end

%% Visualising histograms

f = figure;
subplot(1,2, 1), histogram('BinCounts', Data0_pdf, 'BinEdges', edges0)
ylabel('P(X=x)'), xlabel('x'), title('Data0 pdf')
subplot(1,2, 2), histogram('BinCounts', DataNew_pdf, 'BinEdges', edges1)
ylabel('P(X=x)'), xlabel('x'), title('DataNew pdf')
complete_title = sgtitle(sprintf('Experimental and sample probability distributions (N =↙
%d)',bins));
complete_title.FontSize = 14; complete_title.FontWeight = "bold";
annotation('textbox',[.345 .76 .11 .1],'String',...
    sprintf("DKL = %0.5f", DKL_0))
annotation('textbox',[.785 .76 .11 .1],'String',...
    sprintf("DKL = %0.5f", DKL_New))
end
```