Assignment 3

Ex. 3

Consider the simple linear regression model:

$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, \ldots, N$

where β_0 and β_1 are the unknown parameters. Assume that the ϵ_i's are iid t-distributed data with unknown degrees of freedom ν.

(a) Write the loglikelihood function for the MLE estimation of the three unknowns parameters.

Our model can be written as

$y \mid x \sim \tau(\beta_0 + \beta_1 x, \nu)$

where $\tau(\mu, \eta)$ is a (noncentral) Student's t-distribution centered in μ. The loglikelihood l of our model is

$$l((\beta_0, \beta_1, \nu \mid y, x) = \log \prod_{i=1}^{N} f(y_i - \beta_0 - \beta_1 x_i \mid \nu) = \sum_{i=1}^{N} \log f(y_i - \beta_0 - \beta_1 x_i \mid \nu)$$

where $f(\cdot \mid \nu)$ is the density function of a standard Student's t-distibution with ν degrees of freedom, given by

$$f(t \mid \nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\nu}\Gamma\left(\frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}} = \frac{1}{\sqrt{\nu} B\left(\frac{1}{2}, \frac{\nu}{2}\right)} \left(1 + \frac{t^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

so the log density takes the form

$$l((\beta_0, \beta_1, \nu \mid y, x) = N \log\left(\frac{1}{\sqrt{\nu} B\left(\frac{1}{2}, \frac{\nu}{2}\right)}\right) - \frac{\nu+1}{2} \sum_{i=1}^{N} \log\left(1 + \frac{(y_i - \beta_0 - \beta_1 x_i)^2}{\nu}\right)$$

and the MLE estimators are

$\tilde{\beta}_0, \tilde{\beta}_1, \tilde{\nu} = \text{argmax}_{\beta_0, \beta_1, \nu} l((\beta_0, \beta_1, \nu \mid y, x)$

(b) and (c) Write a function to find the MLE estimates and use the data provided to find the fit

In [146]:
```
function logl(β0, β1, v, x, y)
    N = length(y)
    df_term = N * log(1 / sqrt(v) / beta(0.5, 0.5v))
    error_term = -0.5(v + 1) * sum(log.(1 + ((y - β0 - β1*x)).^2 / v))
    return df_term + error_term
end
```

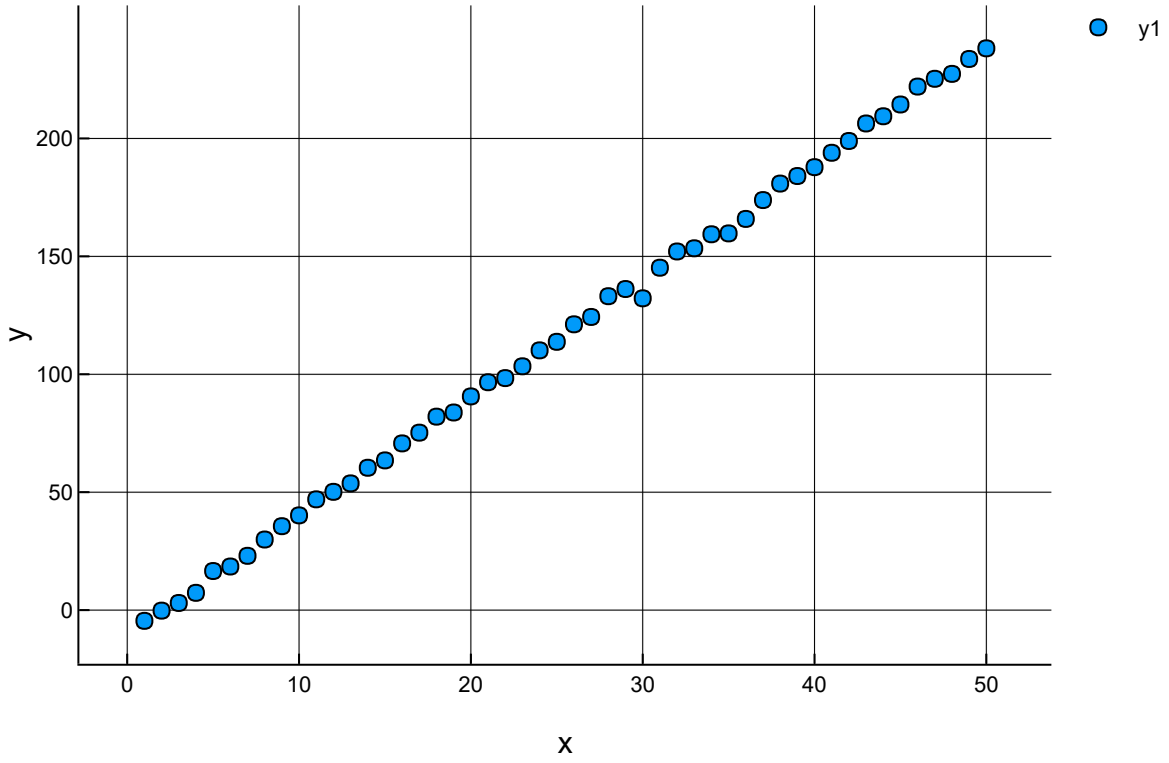
Out[146]:
```
logl (generic function with 2 methods)
```

In [147]:
```
using DataFrames # read the data
using Plots # visualize the data with plots
using StatPlots # visualize the data with plots
```

In [148]:
```
tdata = readtable("../data/tdata.tsv");
```

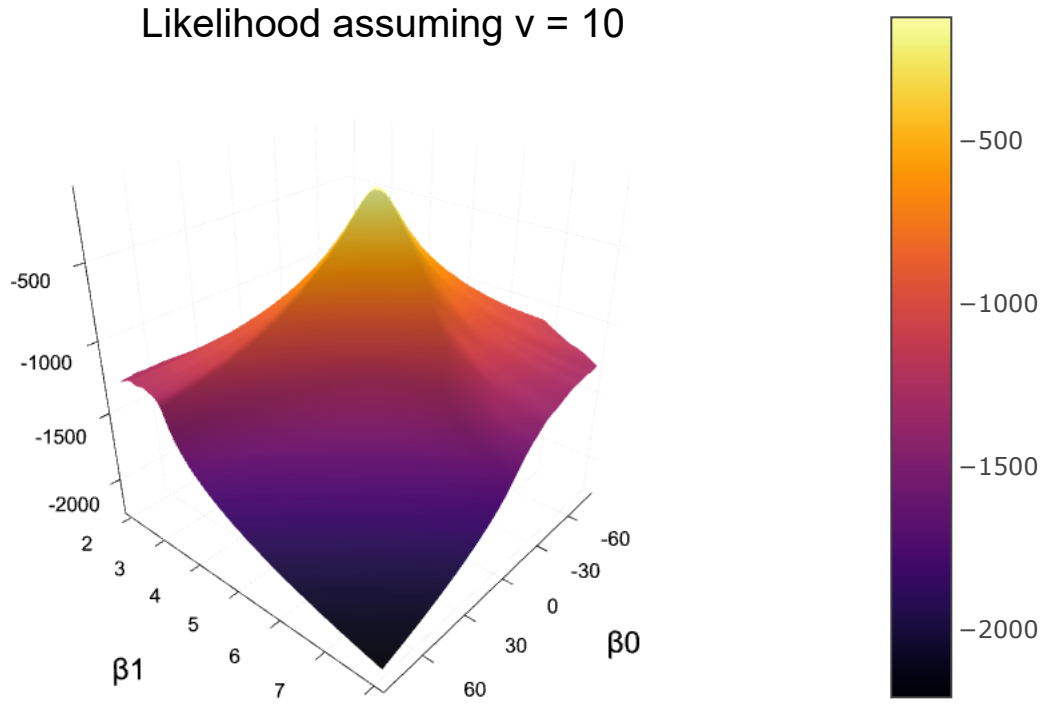
In [149]:
```
scatter(tdata, :x, :y)
```

Out[149]:



In [185]:
```
surface(
    linspace(-80, 80, 100), linspace(2, 8, 100),
    (x, y) -> logl(x, y, 10, tdata[:x], tdata[:y]),
    xlabel = "β0", ylabel = "β1", title = "Likelihood assuming v = 10"
)
```

Out[185]:



In [91]:
```
using NLopt # API to standard non-linear optimizer
```

In [138]:
```
function t_regression_mle(x, y)
    opt = Opt(:LD_MMA, 3)
    f = θ -> -logl(θ[1], θ[2], θ[3], x, y)
    g = Calculus.gradient(f)
    function obj(θ, grad)
        if length(grad) > 0
            grad .= g(θ)
        end
        f(θ)
    end
    lower_bounds!(opt, [-Inf, -Inf, .001])
    ftol_rel!(opt, 1e-15)
    ftol_abs!(opt, 1e-15)
    min_objective!(opt, obj)
    minf, minx, ret = optimize(opt, [0., 0., 1.])
    β0, β1, v = minx
    fitted = β0 + β1*x
    return minx, fitted
end
```

Out[138]: t_regression_mle (generic function with 1 method)

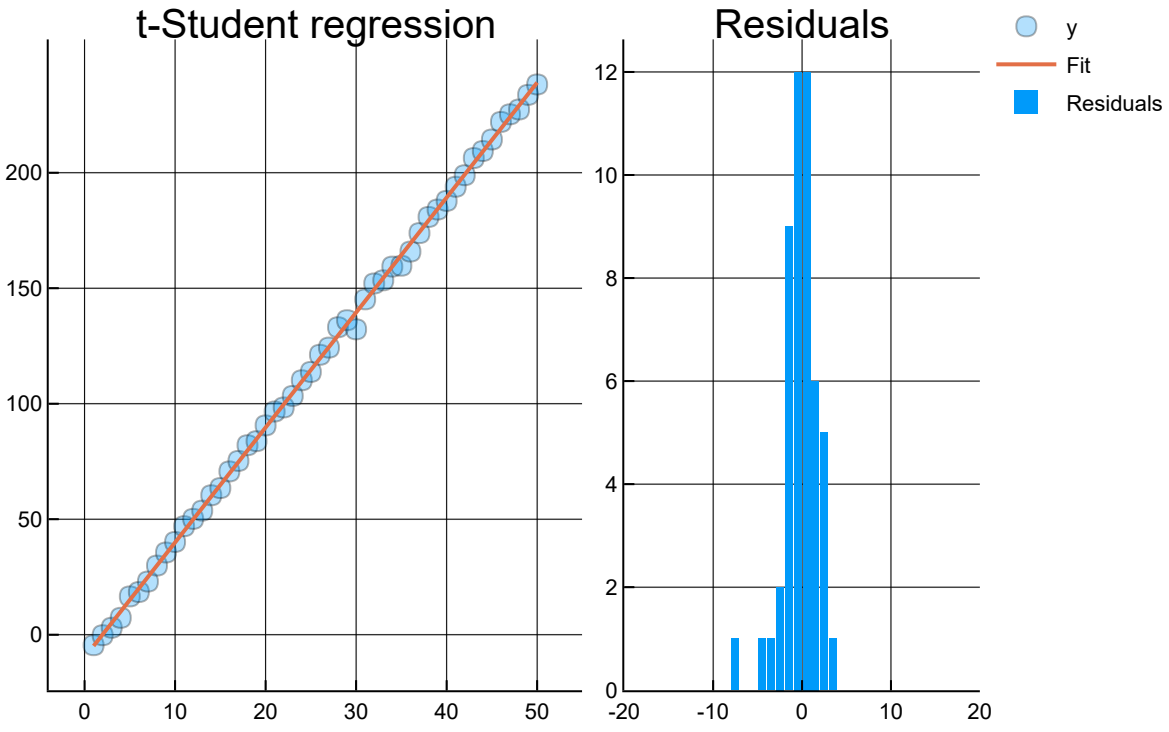
These are the MLE estimates for the t-student fit:

In [155]:
```
tic()
params, yTS =  t_regression_mle(tdata[:x], tdata[:y])
β0TLS, β0TLS, vTS = params
q = toq()
println("MLE Estimates:\n\t[β0, β1, v] = ", params)
println("Solved in: \n\t", round(q, 4), "s")
```

```
MLE Estimates:
        [β0, β1, v] = [-9.83584, 4.97765, 2.2202]
Solved in:
        0.0163s
```

In [140]:
```
resTS = tdata[:y] - yTS
plot(
    plot(tdata[:x], [tdata[:y] yTS],
        seriestype = [:scatter :line], title = "t-Student regression",
        labels = ["y" "Fit"],
        ms = [5 0], alpha = [0.3 1], lw = [1 2]),
    histogram(resTS, title = "Residuals", xlims = (-20,20), labels = "Residuals")
    layout = @layout [a{0.6w} b{0.4w}]
)
```

Out[140]:



(d) Now find and compare with the OLS estimators

In [141]:
```
x = tdata[:x]
y = tdata[:y]
X = [ones(length(x)) x]
β0OLS, β1OLS = Symmetric(X' * X) \ X' * y
```

Out[141]:
```
2-element DataArrays.DataArray{Float64,1}:
 -10.0878
   4.98212
```

We see that the values are similar but not the same

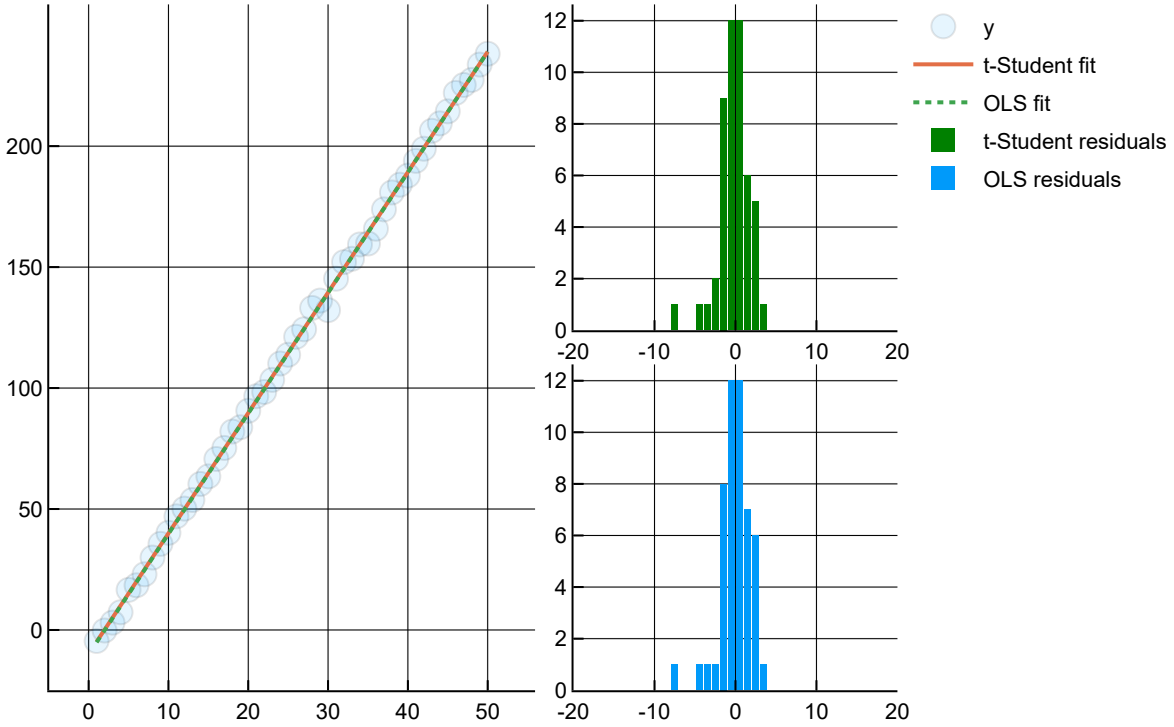
In [144]:
```
[β0OLS, β1OLS] - [β0TS, β1TS]
```

Out[144]:
```
2-element Array{Float64,1}:
 -0.251996
  0.00446813
```

The largest difference is in the incercept. We can visually compare the two models, but they look very similar.

In [145]:
```
yOLS = β0OLS + β1OLS*x
resOLS = y - yOLS
plot(
    plot(x, [y yTS yOLS],
        st = [:scatter :line :line],
        labels = ["y" "t-Student fit" "OLS fit"],
        lw = [1 2 2],
        alpha = [0.1 1 1],
        ms = [6 1 1],
        ls = [:dot :solid]
    ),
    histogram(resTS, xlims = (-20,20), labels = "t-Student residuals", color = :g
    histogram(resOLS, xlims = (-20,20), labels = "OLS residuals"),
    layout = @layout [a{0.6w} [b{0.5h}; c{0.5h}]]
)
```

Out[145]:



(e) Quality of the fit

It is very important to check if we have a good model. Formally, we would like to to a goodness-of-fit test.

In [172]:
```
using Distributions
using HypothesisTests
using KernelDensity
```

In [173]:
```
dt = TDist(vTS)
```

Out[173]:
```
Distributions.TDist{Float64}(v=2.2201984602287608)
```

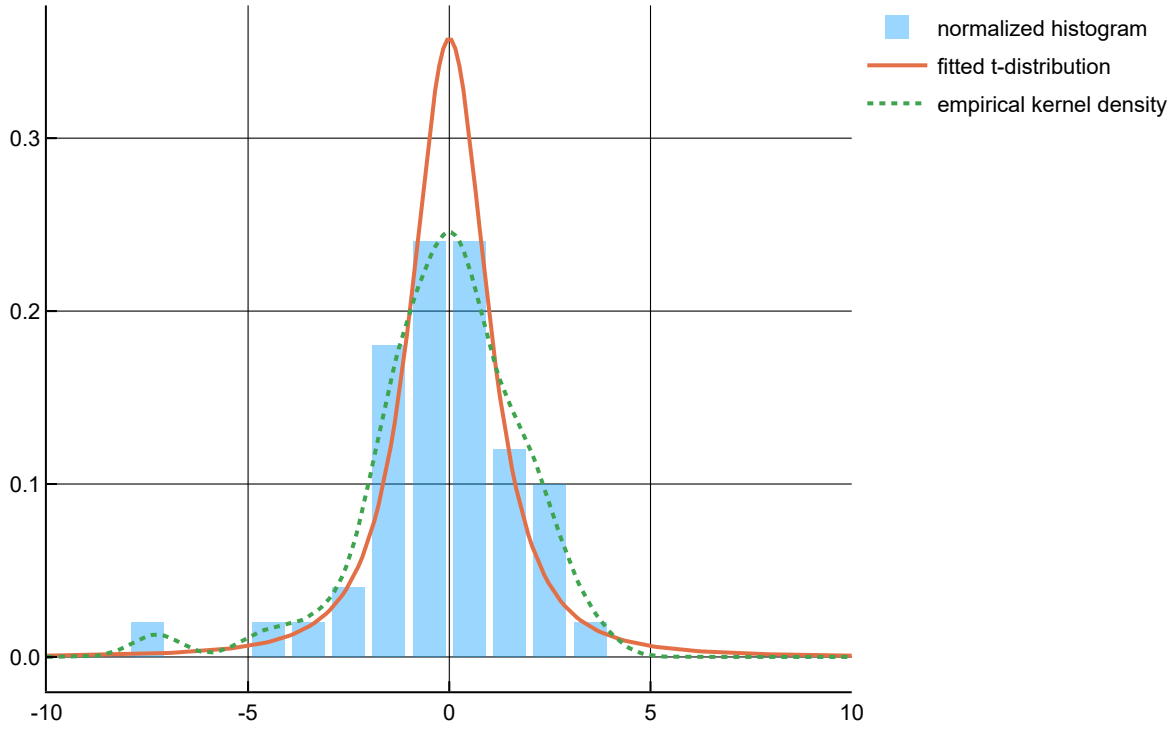
In [174]:
```
HypothesisTests.ExactOneSampleKSTest(resTS, dt)
```

Out[174]:
```
Exact one sample Kolmogorov-Smirnov test
----------------------------------------
Population details:
    parameter of interest:   Supremum of CDF differences
    value under h_0:         0.0
    point estimate:          0.09688307793959064

Test summary:
    outcome with 95% confidence: fail to reject h_0
    two-sided p-value:           0.6995082537077406

Details:
    number of observations:   50
```

The outcome of our goodness-of-fit test says that we cannot reject that residuals are t-distributed with the MLE value we obtained. Which is good! We can also add a plot

In [184]:
```
plot(resTS, st = :histogram, xlims = (-10,10), normed = true, labels = "normalize
plot!(linspace(-10, 10, 200), t -> pdf(dt, t), st = :path, lw = 2, labels = "fitt
plot!(linspace(-10, 10, 200), t -> pdf(kde(resTS), t), st = :path, lw = 2, ls = :
```

Out[184]:



The Kernel Density will tend to underestimate the heavy tails, which the fitted t-distribution shows.