

## qtplot/src/analysis

Class Fit

enthält protected Variable

bool d\_scale\_errors; Fit.h : 251

und public function Fit.h : 133

```
void scaleErrors (bool yes=true)
```

```
{ d_scale_errors = yes; }
```

Variable d\_scale\_errors wird in

```
void Fit::init ()
```

```
d_scale_errors = false Fit.cpp : 94
```

gesch.

\* init wird in Constructor Fit::Fit abgerufen

↳ Defaultmäßig ist d\_scale\_errors = false

scaleErrors wird abgerufen in

```
ApplicationWindow.cpp : 13381 * (1)
```

```
Fitter -> scaleErrors (Fit-scale_errors); ==>
```

↑

class Fit

```
public: bool Fit-scale_errors; ApplicationWindow.cpp : 1310
```

```
Fit-scale_errors = true; ApplicationWindow.cpp : 731
```

QtPlot setzt Vorgabewert auf true

dieser muss vom gewählten Fitter übernommen werden !

\*1

src/core/ApplicationWindow.cpp : 13549

```
if (operation != FitLinear && operation != FitSlope)
```

```
{ fitter -> guessInitialValues();
```

```
fitter -> scaleErrors (fit_scale_errors);
```

```
fitter -> generateFunction (generateUniformFitPoints,  
                           fitPoints);
```

```
} else if (d_2 - Linear - fit - points)
```

```
fitter -> generateFunction (generateUniformFitPoints,
```

LinearFit und LinearSlopeFit übernehmen nicht den Wert aus dem ApplicationWindow.  
Der default Wert false aus Class Fit wird nicht überschrieben.

```
13398: void ApplicationWindow::analyzeCurve
```

```
(Graph *g, OutPlotCurve *C, Analysis operation)
```

```
switch (operation) {
```

```
case FitLinear:
```

```
13353: fitter = new LinearFit(this, g); #
```

```
break;
```

```
case FitSlope:
```

```
14492: fitter = new LinearSlopeFit(this, g); #
```

```
break
```

```
# defined in src/analysis/PolynomialFit.h
```

## SVC/analysis/Polynomial Fit.h

```
65: class LinearFit : public Fit
    {
        void fit();
    private
        void init()
    }
```

```
85: class LinearSlopeFit : public Fit
    {
        void fit();
    private
        void init();
    }
```

## PolynomialFit.cpp

```
272: void LinearFit::init() {
```

```
274:     d_scale_errors = false
```

```
378 void LinearSlopeFit::init() {
```

```
380:     d_scale_errors = false
```





SVC / analysis / Polynomial Fit . epp

293: void LinearFit :: Fit ( ) {

305 ~~if~~ if (d\_weighting == NoWeighting)

gsl\_fit\_linear ( . . . . ) GSL-Referenz p 403

else

gsl\_fit\_wlinear ( . . . . ) GSL-Referenz p 403

weighted data  $\Rightarrow$

~~Variance~~ Covariance Matrix

$$C_{ab} = \sum_i \frac{1}{\omega_i} \frac{\partial c_a}{\partial y_i} \frac{\partial c_b}{\partial y_i} \quad \omega_i = \frac{1}{\sigma_i^2}$$

unweighted data  $\Rightarrow$  Variance - covariance Matrix

$$\omega_i = \frac{1}{\sigma^2} \text{ mit}$$

$$\sigma^2 = \frac{\sum (y_i - Y(c, x_i))^2}{(n-p)}$$

$$C_{ab} = \sigma^2 \underbrace{\sum_i \frac{\partial c_a}{\partial y_i} \frac{\partial c_b}{\partial y_i}}_{\text{Covariance Matrix}}$$

$$\sigma_{c_a} = \sqrt{C_{aa}}$$

$\Rightarrow$  weighted Data : entspricht Scale Errors = No  
unweighted Data : entspricht Scale Errors = Yes  
Option "Scale Errors" ist wirkungslos

Notwendig, da Festlegung durch die Wahl der GSL-Funktion vorgegeben ist. Eine Anwendung von Scale Errors=true auf unweighted Data führt zu falschen Ergebnissen.