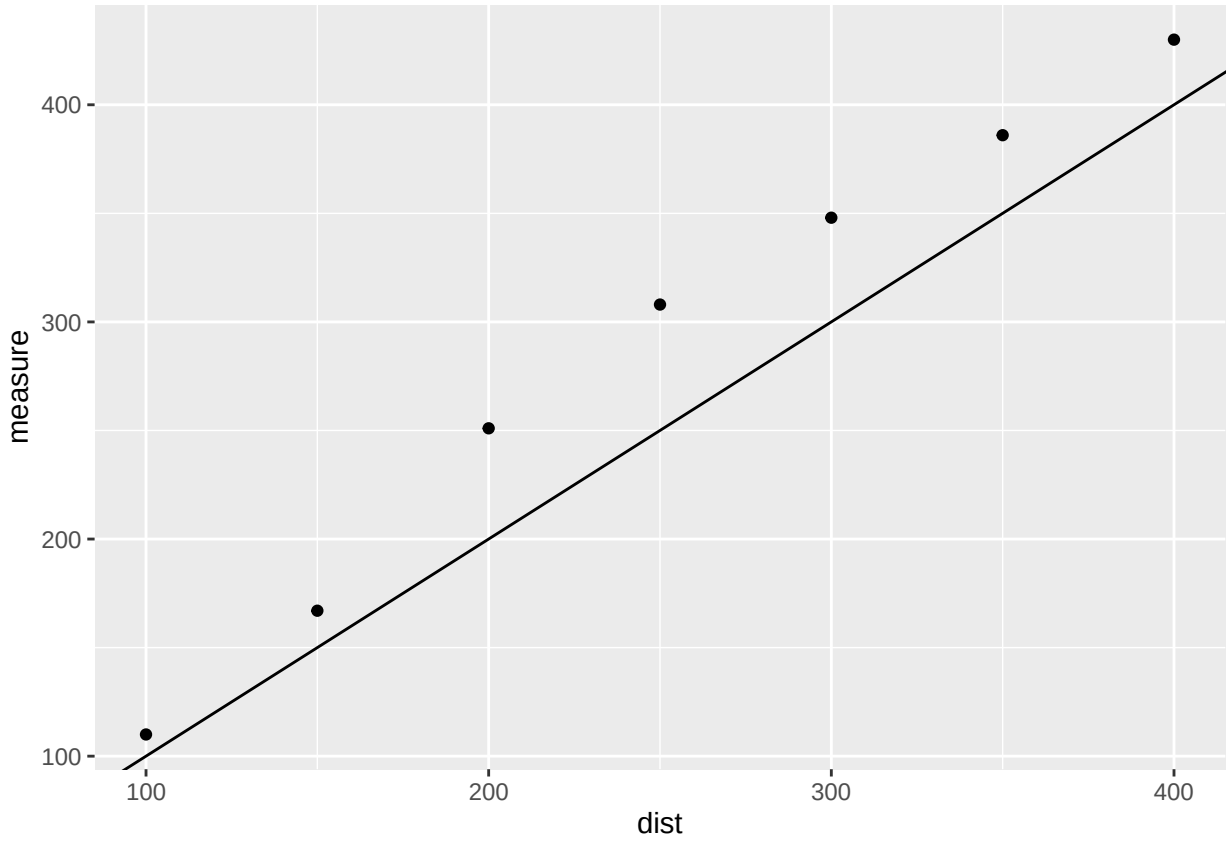
IR Distance Sensor Regression

20212978

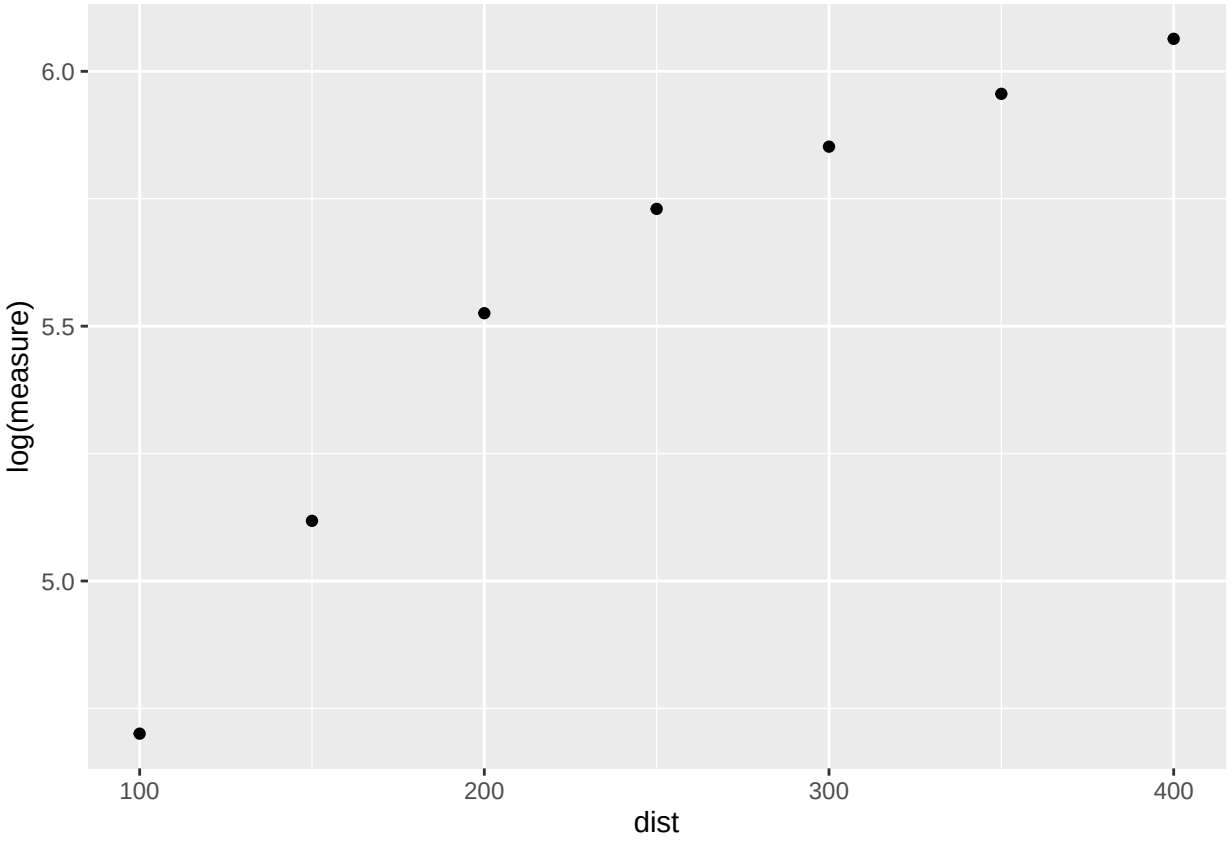
1. Enter observed data:

```
library(ggplot2)
distance = data.frame(list(
  "dist"=c(100,150,200,250,300,350,400),
  "measure"=c(110,167,251,308,348,386,430)
))
ggplot(distance, aes(x=dist, y=measure)) + geom_point() + geom_abline()
```



2. Try log transformation

```
ggplot(distance, aes(x=dist, y=log(measure))) + geom_point() + geom_abline()
```



3. Run non-linear least squares for simple log model

$$f(dist) = \log(measure)$$
$$f(dist) = multiplier \times \log(dist - x_{offset}) + y_{offset}$$

```
b1 <- nls(log(measure) ~ multiplier * log(dist - x_offset) + y_offset, data=distance)
```

```
## Warning in nls(log(measure) ~ multiplier * log(dist - x_offset) + y_offset, : No starting values spe
## Initializing 'multiplier', 'x_offset', 'y_offset' to '1.'.
## Consider specifying 'start' or using a selfStart model
```

```
b1
```

```
## Nonlinear regression model
##   model: log(measure) ~ multiplier * log(dist - x_offset) + y_offset
##    data: distance
## multiplier   x_offset   y_offset
##     0.7051    52.0695     1.9576
##  residual sum-of-squares: 0.01012
##
## Number of iterations to convergence: 8
## Achieved convergence tolerance: 1.931e-06
```

4. solve for inverse

$$f(dist) = \log(measure)$$
$$dist = f^{-1}(\log(measure))$$

find f^{-1}

$$f(x) = multiplier \times \ln(x - x_{offset}) + y_{offset}$$
$$f^{-1}(x) = \exp(\frac{x - y_{offset}}{multiplier}) + x_{offset}$$
$$\therefore f^{-1}(\ln\ measure) = dist$$

$$ ## 5. check result

```
vals <- as.numeric(unlist(distance["measure"]))
distance_fit = data.frame(list(
  "measure"=as.numeric(unlist(distance["measure"])),
  "fit"= exp((log(vals) - 1.9576)/0.7051) + 52.0695
))
ggplot() + geom_point(aes(x=measure, y=fit), data = distance_fit, color="red")
```