SF2930 - Regression analysis

KTH Royal Institute of Technology, Stockholm

Lecture 6 – Model assesment (MPV 4-5)

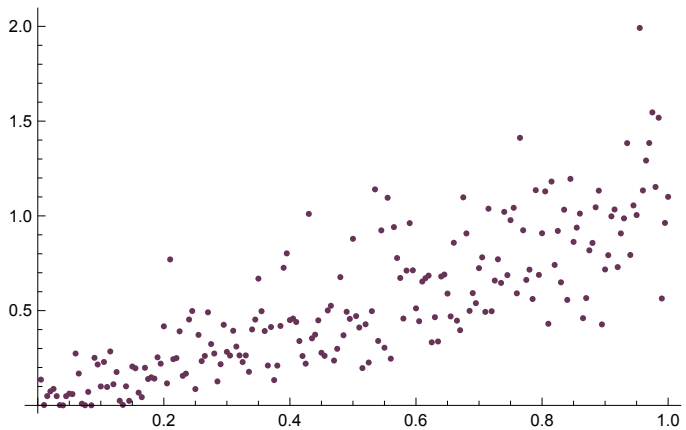
March 13, 2022

Todays lecture

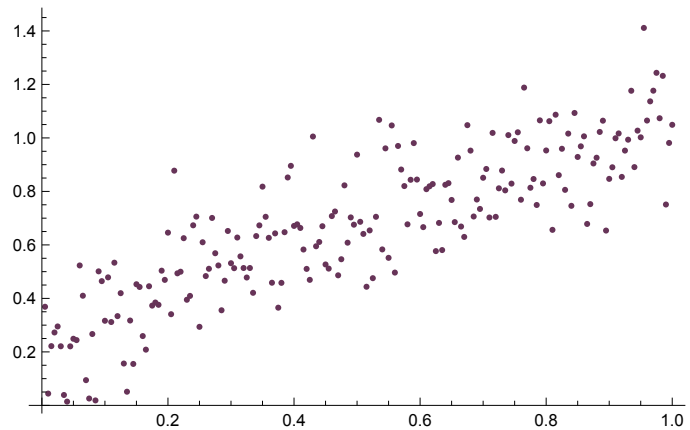
- Variance stabilizing transforms
- The Box-Cox method
- Transformations which linearize a model
- The Box-Hastings algorithm

Variance-stabilizing transformations

If σ^2 does not appear constant, we can sometimes solve this by applying a transform of the response variable y before applying a linear model.



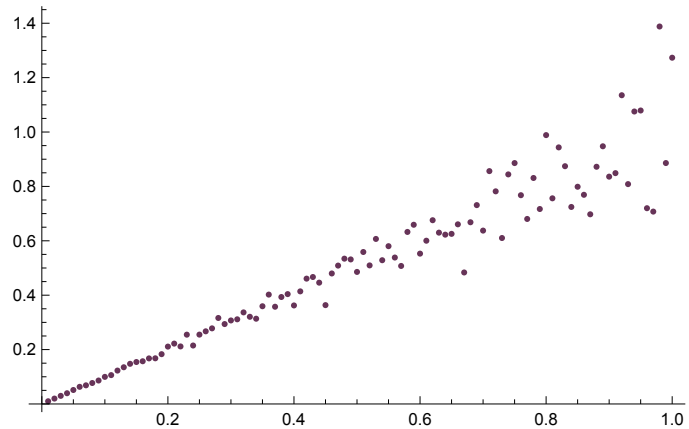
A sample $((x_i, y_i))$ before rescaling.



The same sample after the transform $y_i \mapsto \sqrt{y_i}$.

In this setting, we might consider the model $\sqrt{y_i} = \beta_0 + \beta_1 x_i + \varepsilon_i$ instead.

Variance-stabilizing transformations



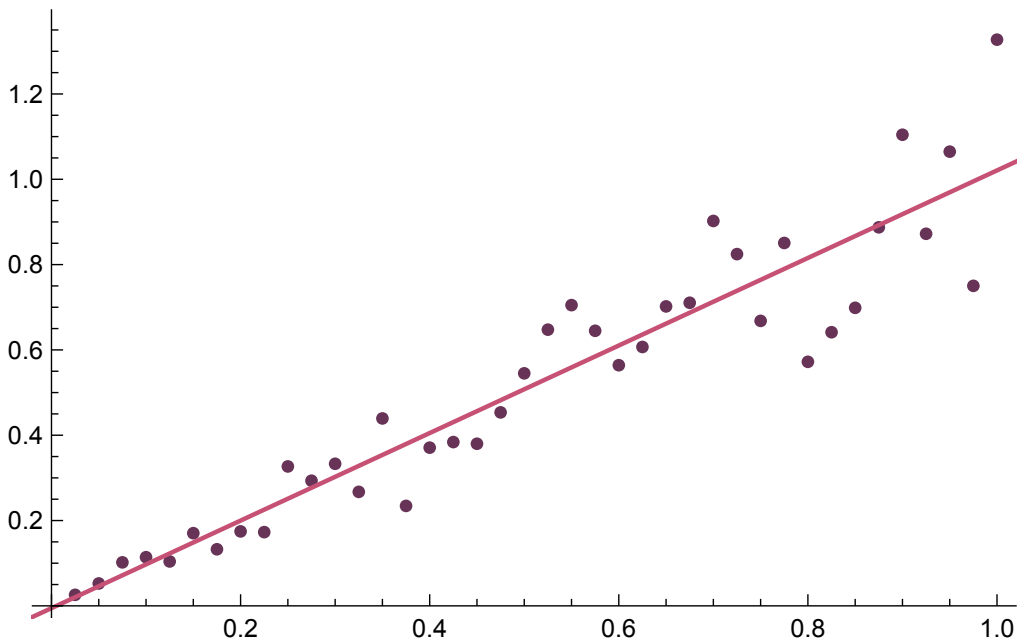
How can we find a suitable transformation of the data?

1. By inspection
2. By understanding the data.
3. By using the Box-Cox method.

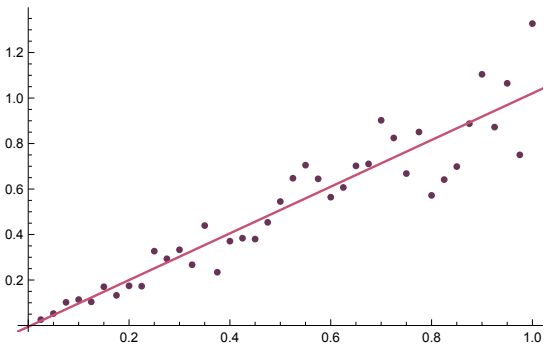
Transformations of y

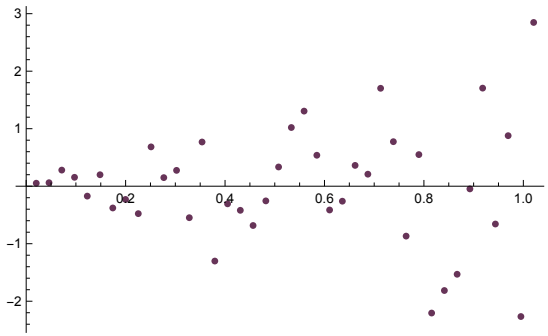
Pattern in σ^2	Transformation
$\sigma^2 \propto 1$	$y' = y$
$\sigma^2 \propto \mathbb{E}[y \mid x]$	$y' = \sqrt{y}$
$\sigma^2 \propto \mathbb{E}[y \mid x](1 - \mathbb{E}[y \mid x])$	$y' = \arcsin(\sqrt{y})$
$\sigma^2 \propto \mathbb{E}[y \mid x]^2$	$y' = \log(y)$
$\sigma^2 \propto \mathbb{E}[y \mid x]^3$	$y' = y^{-1/2}$
$\sigma^2 \propto \mathbb{E}[y \mid x]^4$	$y' = y^{-1}$

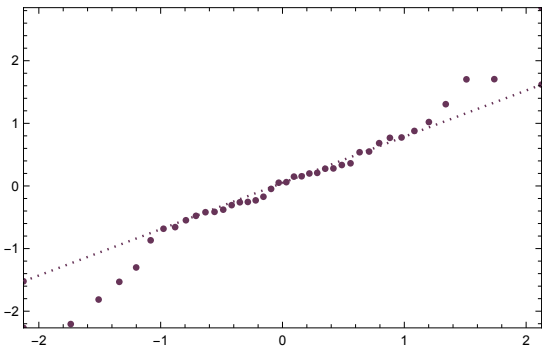
Table 5.1 in MPV

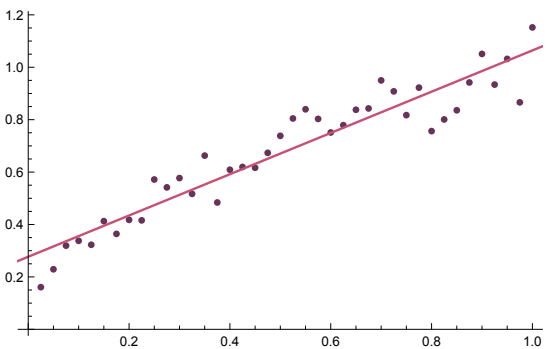


Example: $\sigma^2 \propto \mathbb{E}[y \mid x]$

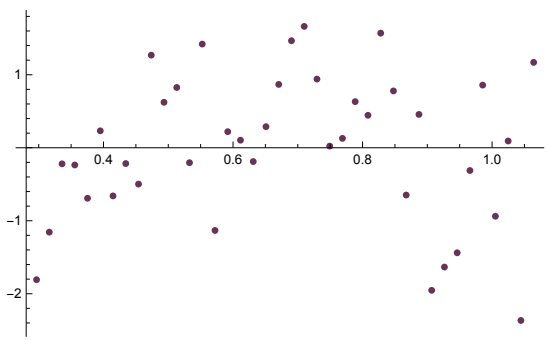


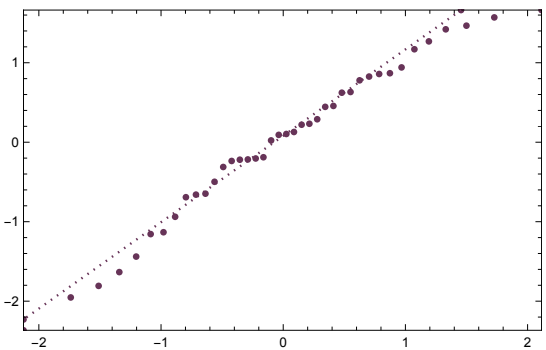




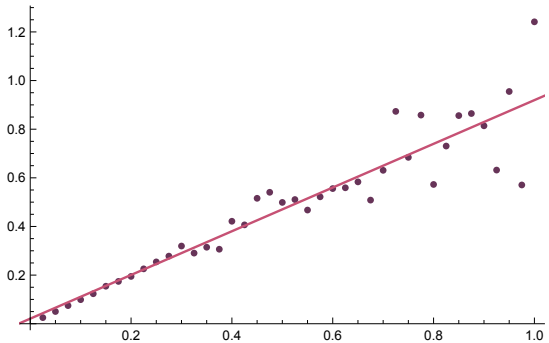


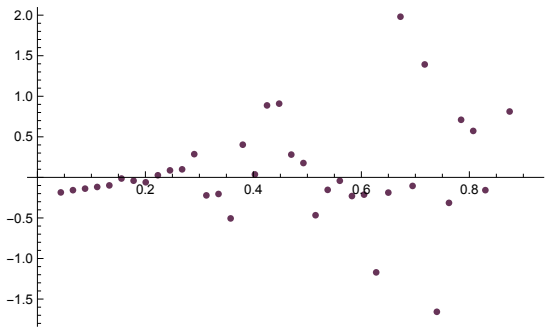
$y' = \sqrt{y}$

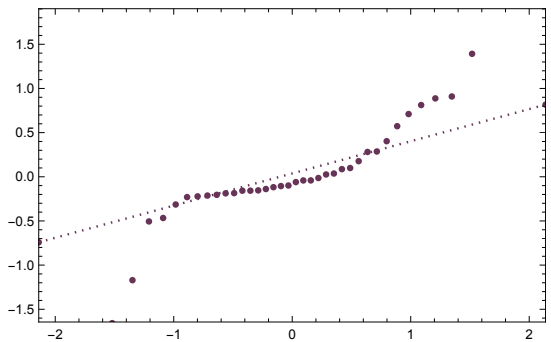


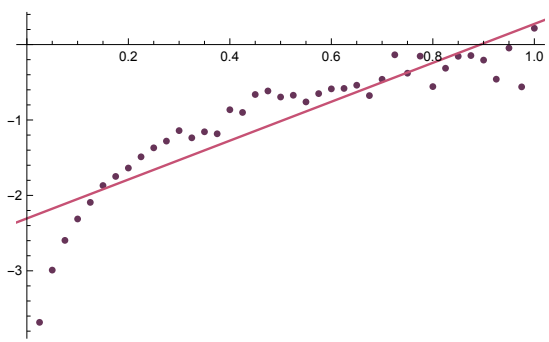


Example: $\sigma^2 \propto \mathbb{E}[y \mid x]^2$

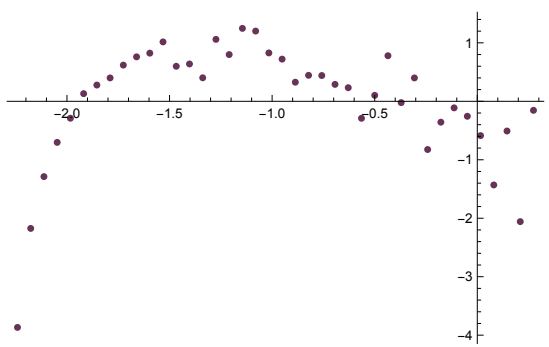


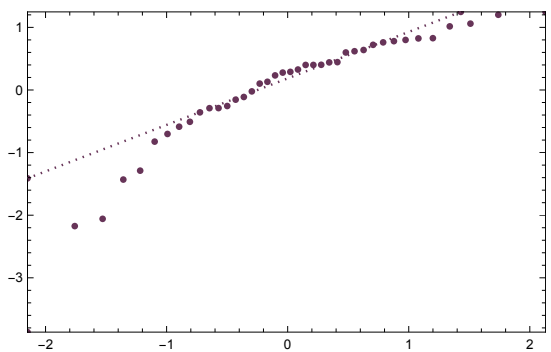






$$y' = \log y$$





The Box-Cox method, version 1

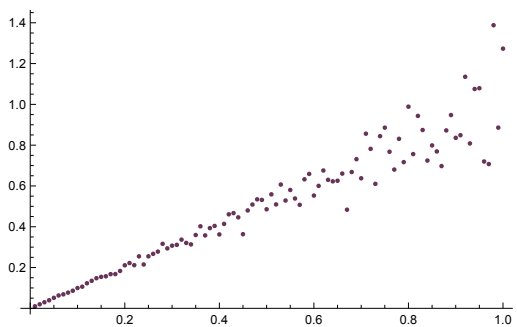
Model

$$y^{\lambda} = f(x) + \varepsilon$$

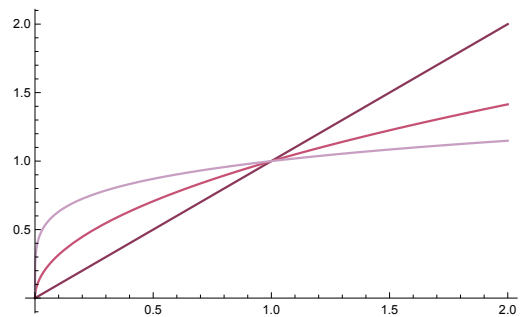
How can we choose the "best" λ?

Idea 1

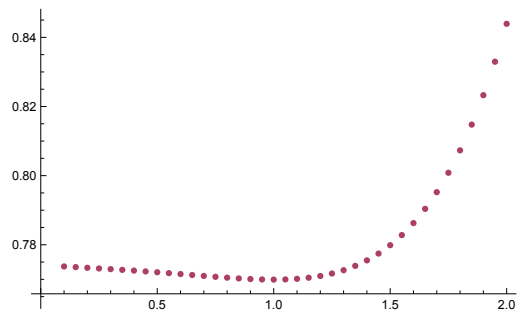
For some subset of all realistic λ, pick the λ which minimizes $SS_{Res}(y^{\lambda})$.



A sample $((x_i, y_i))$.



(x, x^{λ})



$(\lambda, SS_{Res}(y^{\lambda}))$

!! When λ is close to 0, then all y-values will be very close.

The Box-Cox method, version 2

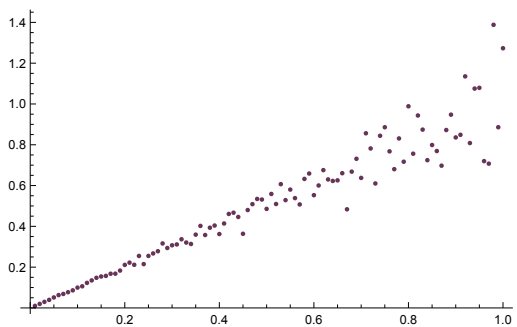
Model

$$y^\lambda = f(x) + \varepsilon$$

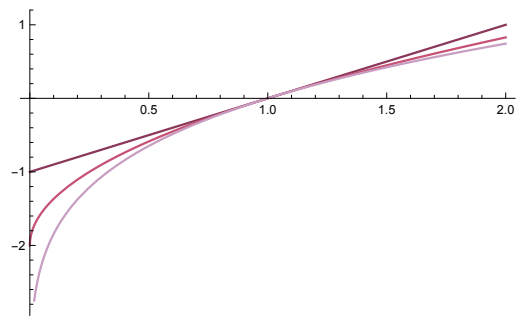
How can we choose the "best" λ?

Idea 2

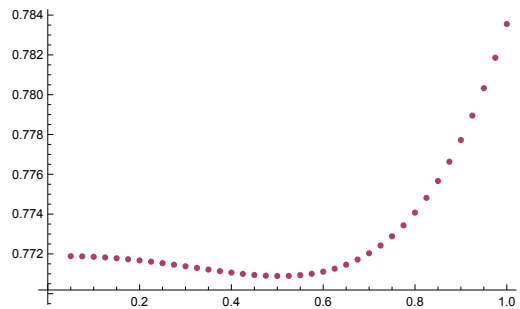
For some subset of all realistic λ, pick the λ that minimizes $SS_{Res}((y^\lambda - 1)/\lambda)$.



A sample $((x_i, y_i))$.



$(x, (x^\lambda - 1)/\lambda)$



$(\lambda, SS_{Res}((y^\lambda - 1)/\lambda))$

!! The function $(y^\lambda - 1)/\lambda$ has a lot of variation in λ, which makes it difficult to compare the residuals.

The Box-Cox method, version 3

Model

$$y^{\lambda} = f(x) + \varepsilon$$

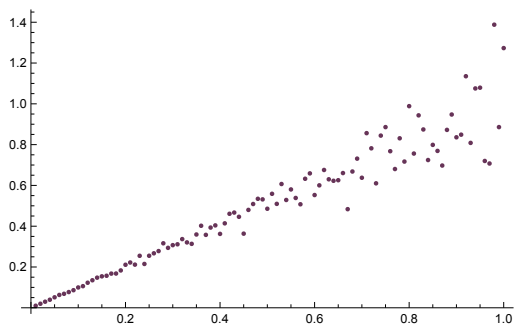
How can we choose the "best" λ?

Idea 3

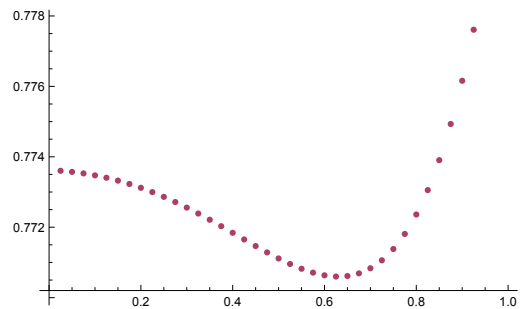
Let $\dot{y} := \sqrt[n]{\prod y_i}$ and define

$$y \mapsto y^{(\lambda)} := \begin{cases} \frac{y^{\lambda}-1}{\lambda \dot{y}^{\lambda-1}} & \text{if } \lambda \neq 0 \\ \dot{y} \log y & \text{if } \lambda = 0. \end{cases}$$

For some subset of all realistic λ, pick the λ that minimizes $SS_{Res}(y^{(\lambda)})$.



A sample $((x_i, y_i))$.



$(\lambda, SS_{Res}(y^{(\lambda)}))$

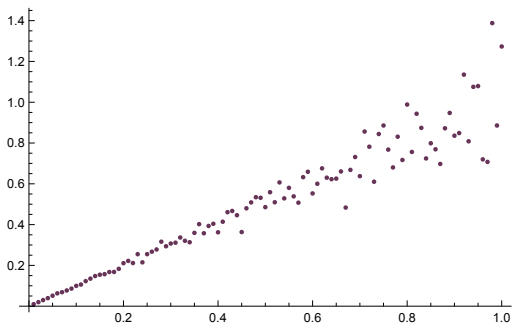
With this scaling, the residuals are comparable.

The Box-Cox method

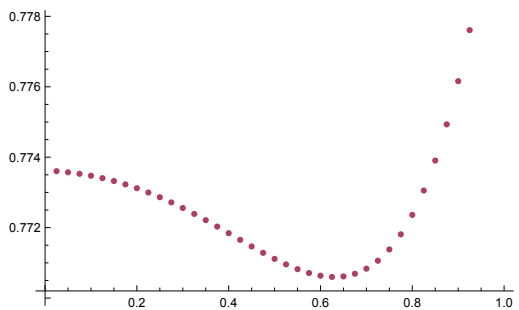
$$\dot{y} := \sqrt[n]{\prod y_i} \qquad \text{and} \qquad y^{(\lambda)} := \begin{cases} \frac{y^\lambda - 1}{\lambda \dot{y}^{\lambda-1}} & \text{if } \lambda \neq 0 \\ \dot{y} \log y & \text{if } \lambda = 0. \end{cases}$$

Algorithm

1. Assume that the "correct" model is given by $y^\lambda = f(x) + \varepsilon$.
2. Pick some finite set of values for λ.
3. Calculate $SS_{Res}(y^{(\lambda)})$ for each λ in the chosen set.
4. Pick, approximately, the λ which minimizes $SS_{Res}(y^{(\lambda)})$.
5. Use the transform $y \mapsto y^\lambda$ (note, not $y \mapsto y^{(\lambda)}$).



A sample $\big((x_i, y_i)\big)$.



$\big(\lambda, SS_{Res}(y^{(\lambda)})\big)$

The Box-Cox method and MLE

$$\dot{y} := \sqrt[n]{\prod y_i} \qquad \text{and} \qquad y^{(\lambda)} := \begin{cases} \frac{y^\lambda - 1}{\lambda \dot{y}^{\lambda-1}} & \text{if } \lambda \neq 0 \\ \dot{y} \log y & \text{if } \lambda = 0. \end{cases}$$

MLE estimation of λ

Assume that $\mathbf{y}^{(\lambda)} \sim N(X\boldsymbol{\beta}, \sigma^2 I)$ (parameters here are $\boldsymbol{\beta}$, λ, and σ^2), so that the density for $\mathbf{y}^{(\lambda)}$ is

$$f(\mathbf{y}^{(\lambda)}) = e^{-(\mathbf{y}^{(\lambda)} - X\boldsymbol{\beta})^T (\mathbf{y}^{(\lambda)} - X\boldsymbol{\beta})/2\sigma^2} / (2\pi\sigma^2)^{n/2}.$$

Then the likelihood function is given by

$$L(\lambda, \boldsymbol{\beta}, \sigma^2 \mid \mathbf{y}, X) = f(\mathbf{y}^{(\lambda)}) / J(\lambda, \mathbf{y})$$

where $J(\lambda, \mathbf{y}) \approx \dot{y}$ is the Jacobian of the transformation form $\mathbf{y}$ to $\mathbf{y}^{(\lambda)}$.
The MLE for $\boldsymbol{\beta}$ is $\tilde{\boldsymbol{\beta}} := (X^T X)^{-1} X^T \mathbf{y}^{(\lambda)}$ and the MLE for σ^2 is given by $\tilde{\sigma}^2 := \mathbf{y}^{(\lambda)^T} (I - H)\mathbf{y}^{(\lambda)}/n$. Hence

$$\begin{aligned} & l(\lambda \mid \tilde{\boldsymbol{\beta}}(\lambda, X, \mathbf{y}), \tilde{\sigma}^2(\lambda, X, \mathbf{y}), \mathbf{y}, X) \\ & \quad = -(\mathbf{y}^{(\lambda)} - X\tilde{\boldsymbol{\beta}})^T (\mathbf{y}^{(\lambda)} - X\tilde{\boldsymbol{\beta}})/2\tilde{\sigma}^2 - \frac{n}{2} \log(2\pi\tilde{\sigma}^2) - \log J(\lambda, \mathbf{y}) \\ & \quad = -\frac{1}{2} - \frac{n}{2} \log(2\pi\tilde{\sigma}^2(\lambda)) - \log \underbrace{J(\lambda, \mathbf{y})}_{\approx \dot{y}} \approx C - \frac{n}{2} \log SS_{Res}(\lambda/n) \end{aligned}$$

$\rightarrow$ If we minimize $SS_{Res}(y^{(\lambda)})$ we find an approximate MLE estimate of λ.

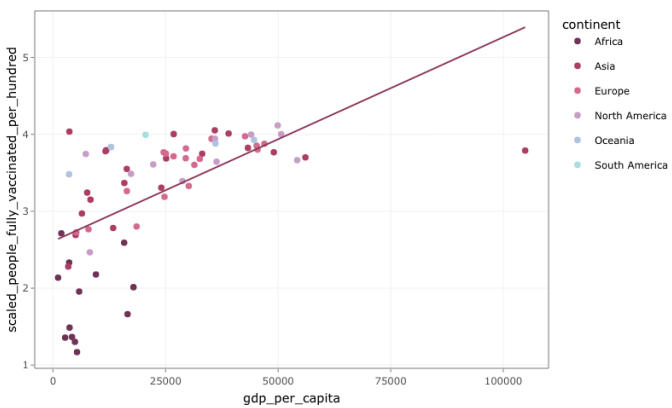
Example

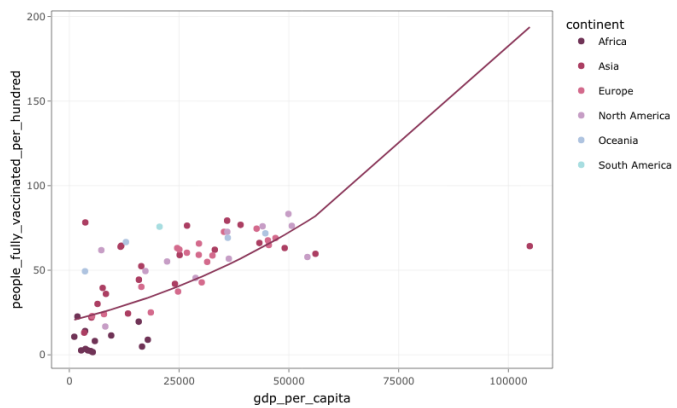
```r
library(AID)

out <- boxcoxnc(df01$people_fully_vaccinated_per_hundred,
    df01$gdp_per_capita, method = "mle", lambda = seq
    (-2,2,0.0001), verbose = T, plot = F)
```

```
Box-Cox power transformation
---------------------------------------------------------------
data : df01$people_fully_vaccinated_per_hundred
lambda.hat : 0.322

Shapiro-Wilk normality test for transf. data (alpha=0.05)
---------------------------------------------------------------
statistic   : 0.9677861
p.value     : 0.07562877
Result      : Transformed data are normal.
---------------------------------------------------------------
```

Example

```
lambda <- .32
df01$scaled_people_fully_vaccinated_per_hundred <- df01$
    people_fully_vaccinated_per_hundred^lambda

df01.model0 <- lm(scaled_people_fully_vaccinated_per_hundred
    ~gdp_per_capita, data = df01)
```





Example

```r
summary(df01.model0)
```

```
Call:
lm(formula = scaled_people_fully_vaccinated_per_hundred ~
    gdp_per_capita, data = df01)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6049 -0.3148  0.1016  0.4015  1.3307

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     2.608e+00  1.284e-01   20.31  < 2e-16 ***
gdp_per_capita 2.656e-05  4.250e-06    6.25 3.41e-08 ***
---
Signif.codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.648 on 66 degrees of freedom
Multiple R-squared:  0.3718,^^IAdjusted R-squared:  0.3623
F-statistic: 39.06 on 1 and 66 DF,  p-value: 3.413e-08
```

Transforming y

Comment

- Both the above methods aim to find λ such that the residuals resulting from fitting the model $y^\lambda = f(x) + \varepsilon$ is as small as possible. However, as we saw above, when we change λ, it is likely that the model should have changed as well (see e.g. the covid data!). Ideally, we should use $f(x)$ which corresponds to the model at the "best" λ, but this is not really possible to do..

Transformations which linearize a model

Sometimes the residuals or other plots suggests that a linear model is not the best possible model for the data. In this case, we can sometimes make a more appropriate model which we then transform to obtain a linear model. Models for which this is possible are called *intrinsically linear*.

Logarithmic transformation

$$y = \beta_0 e^{\beta_1 x + \varepsilon} \varepsilon \mapsto \log y = \log \beta_0 + \beta_1 x + \varepsilon.$$

Note that for the latter model to be a linear model, the error must be in the exponent of the original model.

Reciprocal transformation

$$y = \beta_0 + \beta_1 \underbrace{e^x}_{x'} + \varepsilon \mapsto \log y = \log \beta_0 + \beta_1 x' + \varepsilon.$$

Example: A logarithmic transformation

Model

$$y = e^a x^b$$

Transformed linear model

$$\log y = a + b \log x.$$

```
df01$scaled_gdp_per_capita <- log(df01$gdp_per_capita)
df01$scaled_people_fully_vaccinated_per_hundred <- log(df01$
    people_fully_vaccinated_per_hundred)

df01.model3 <- lm(scaled_people_fully_vaccinated_per_hundred
    ~scaled_gdp_per_capita, data = df01)
```

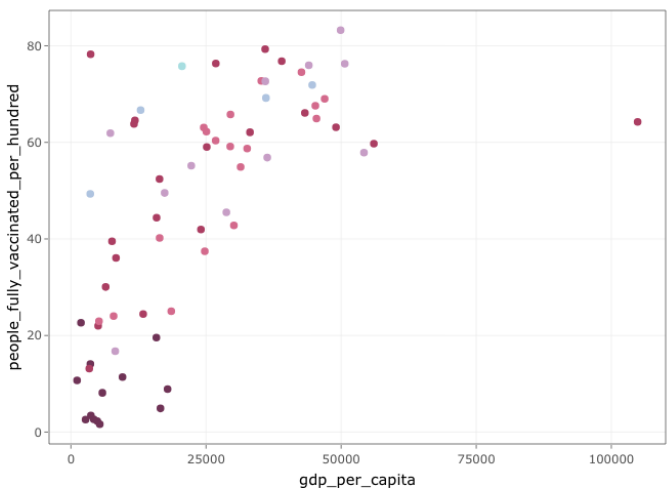
Example: A logarithmic transformation

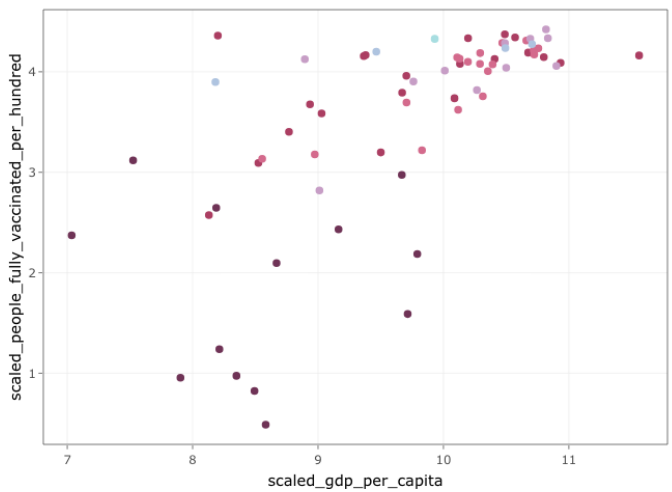
Model

$$y = e^a x^b$$

Transformed linear model

$$\log y = a + b \log x.$$





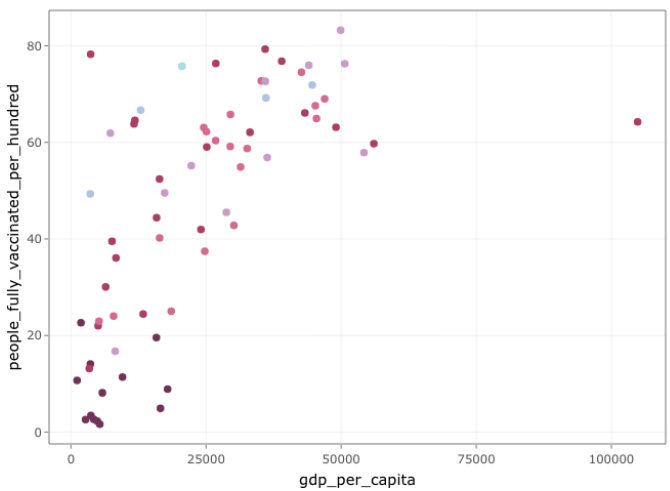
Example: A logarithmic transformation

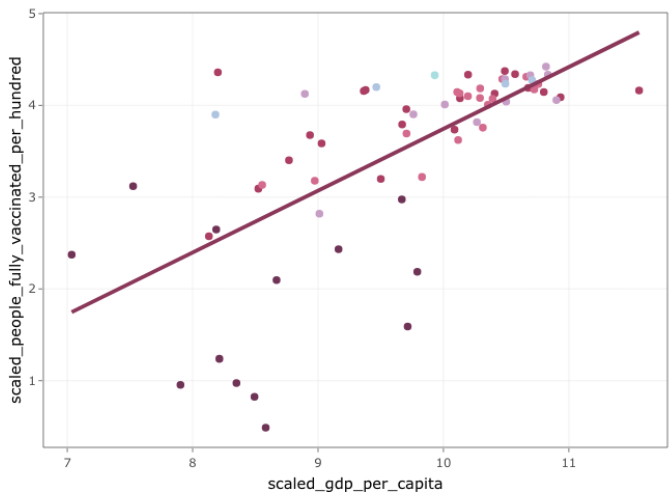
Model

$$y = e^a x^b$$

Transformed linear model

$$\log y = a + b \log x.$$





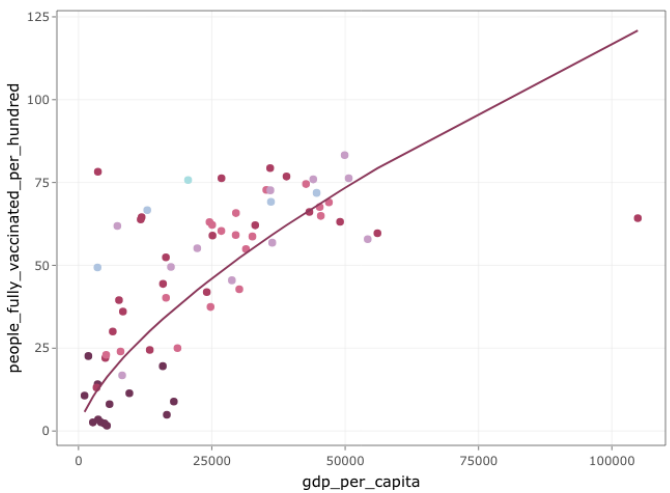
Example: A logarithmic transformation

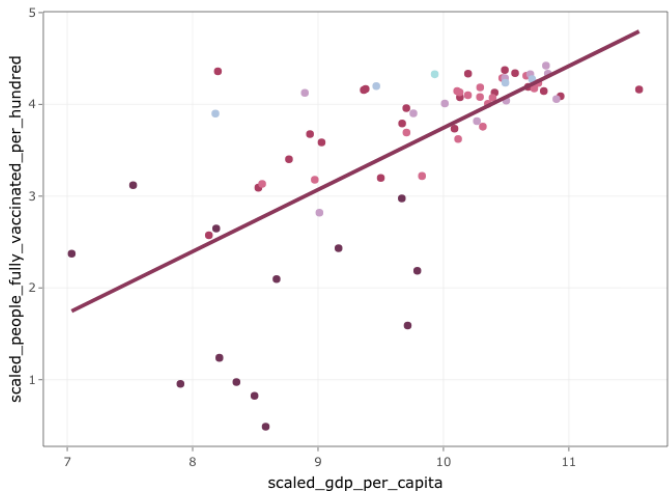
Model

$$y = e^a x^b$$

Transformed linear model

$$\log y = a + b \log x.$$





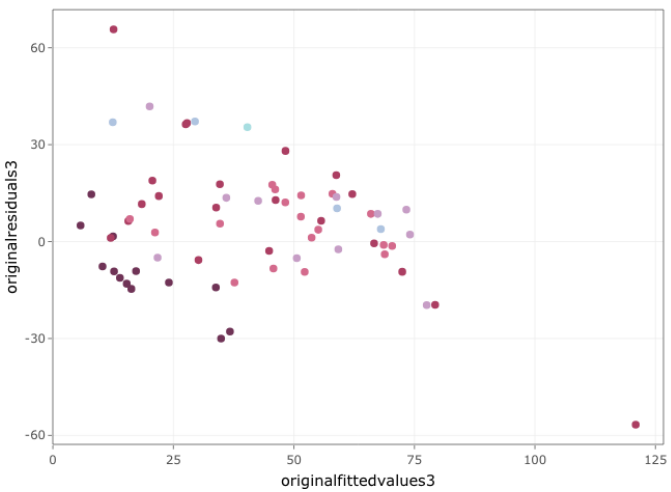
Example: A logarithmic transformation

Model

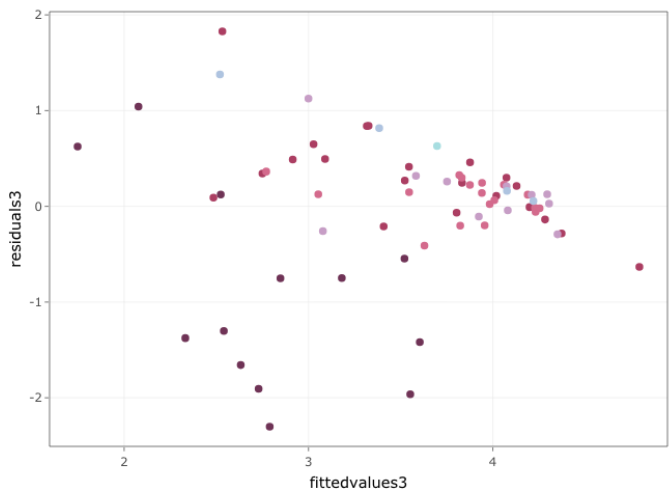
$$y = e^a x^b$$

Transformed linear model

$$\log y = a + b \log x.$$



These residuals are much larger than the residuals in the rescaled model (too large?).



Note that these are the residuals we have minimized

How can we choose the "correct power" for a regressor?

Model

$$y = \beta_0 + \beta_1 x^{\alpha} + \varepsilon$$

How can we find the correct/best α?

Notation

Let

$$\xi(a) := \begin{cases} x^a & \text{if } a \neq 0 \\ \log x & \text{if } a = 0, \end{cases}$$

$\xi := \xi(\alpha)$ and $\xi_0 = \xi(\alpha_0)$.

Then $\mathbb{E}[y \mid x] = \beta_0 + \beta_1 \xi(\alpha)$ for some unknown α. Let

$$f(\xi, \beta_0, \beta_1) := \beta_0 + \beta_1 \xi(\alpha).$$

How can we choose the "correct power" for a regressor?

Notation

$$\xi(a) := \begin{cases} x^a & \text{if } a \neq 0 \\ \log x & \text{if } a = 0, \end{cases} \qquad \xi := \xi(\alpha) \qquad \xi_0 = \xi(\alpha_0).$$

$$f(\xi, \beta_0, \beta_1) := \beta_0 + \beta_1 \xi(\alpha).$$

Main observation

$$\begin{aligned} & f(\xi, \beta_0, \beta_1) \\ & \quad = f(\xi_0, \beta_0, \beta_1) + \big(f(\xi, \beta_0, \beta_1) - f(\xi_0, \beta_0, \beta_1)\big) \\ & \quad \approx f(\xi_0, \beta_0, \beta_1) + (\alpha - \alpha_0) \frac{df(\xi, \beta_0, \beta_1)}{d\alpha}\bigg|_{\xi=\xi_0,\, \alpha=\alpha_0} \\ & \quad = f(\xi_0, \beta_0, \beta_1) + (\alpha - \alpha_0) \frac{df(\xi, \beta_0, \beta_1)}{d\xi}\bigg|_{\xi=\xi_0,} \frac{d\xi}{d\alpha}\bigg|_{\alpha=\alpha_0} \\ & \quad = \beta_0 + \beta_1 \xi_0 + (\alpha - \alpha_0) \beta_1 x^{\alpha_0} \log x. \end{aligned}$$

How can we choose the "correct power" for a regressor?

Notation

$$\xi(a) := \begin{cases} x^a & \text{if } a \neq 0 \\ \log x & \text{if } a = 0, \end{cases} \qquad \xi := \xi(\alpha) \qquad \xi_0 = \xi(\alpha_0).$$

$$f(\xi, \beta_0, \beta_1) := \beta_0 + \beta_1 \xi(\alpha).$$

Main observation

$$f(\xi, \beta_0, \beta_1) = \mathbb{E}[y \mid x] = \beta_0 + \beta_1 \xi^\alpha \approx \beta_0 + \beta_1 \xi_0 + (\alpha - \alpha_0)\beta_1 x \log x.$$

Algorithm

1. Let α_0 be an initial guess for α, and let $\xi_0 := \xi(\alpha_0)$.
2. Calculate the LSE $\hat{\beta}_0$ and $\hat{\beta}_1$ for the equation $\mathbb{E}[x] = \beta_0 + \beta_1 \xi_0$.
3. Calculate the LSE $\hat{\beta}_0^*$, $\hat{\beta}_1^*$, and $\hat{\beta}_2^*$ for the equation $\mathbb{E}[y] = \beta_0 + \beta_1 \xi_0 + \beta_2 x^{\alpha_0} \log x$ (linear in $\boldsymbol{\beta}$).

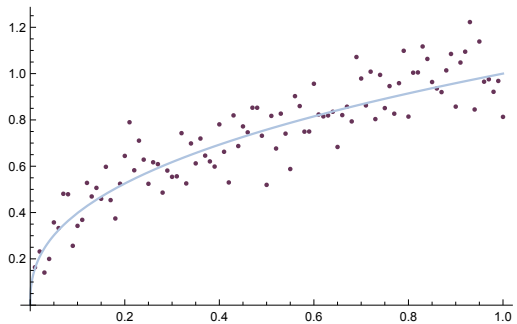
   [By the main observation, we should have $\hat{\beta}_2^* \approx (\alpha - \alpha_0)\beta_1 \approx (\alpha - \alpha_0)\hat{\beta}_1$, and hence $\alpha \approx \alpha_0 + \hat{\beta}_2^*/\hat{\beta}_1$.]

4. Repeat the above steps with the new guess $\alpha_1 := \alpha_0 + \hat{\beta}_2^*/\hat{\beta}_1$.

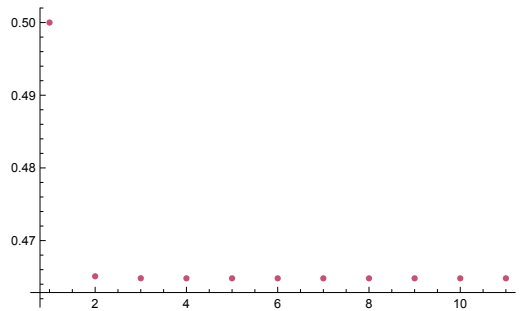
How can we choose the "correct power" for a regressor?

Algorithm

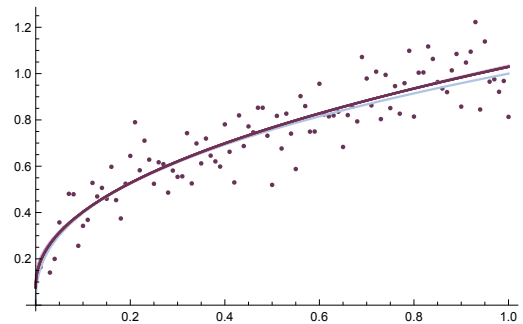
1. Let α_0 be an initial guess for α, and let $\xi_0 := \xi(\alpha_0)$.
2. Calculate the LSE $\hat{\beta}_0$ and $\hat{\beta}_1$ for the equation $\mathbb{E}[x] = \beta_0 + \beta_1 \xi_0$.
3. Calculate the LSE $\hat{\beta}_0^*$, $\hat{\beta}_1^*$, and $\hat{\beta}_2^*$ for the equation $\mathbb{E}[y] = \beta_0 + \beta_1 \xi_0 + \beta_2 x^{\alpha_0} \log x$.
4. Repeat the above steps with the new guess $\alpha_1 := \alpha_0 + \hat{\beta}_2^* / \hat{\beta}_1$.



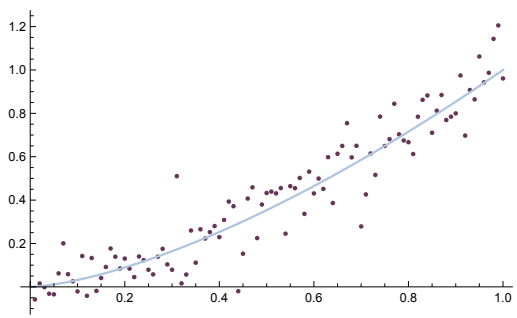
A sample from the model $y = x^{.4} + \varepsilon$.



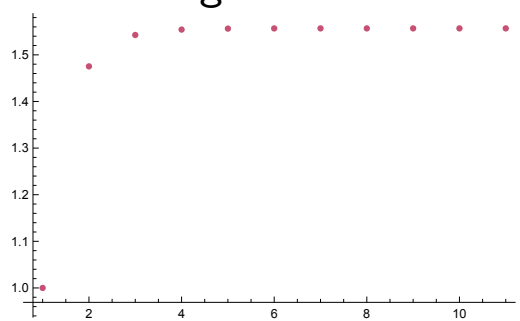
The values of α considered by the algorithm.



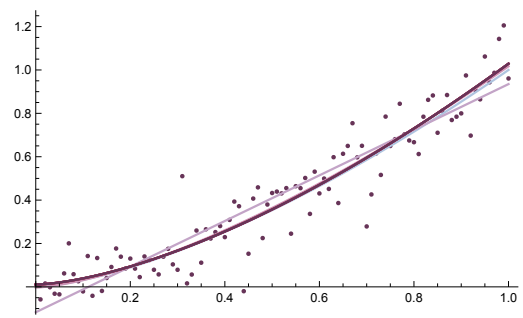
The fits considered by the algorithm.



A sample from the model $y = x^{1.5} + \varepsilon$.



The values of α considered by the algorithm.



The fits considered by the algorithm.

How can we choose the "correct power" for a regressor?

Comments

- It is not at all obvious that this model returns the power which minimizes the least squared error or maximizes any other measure of how good the final model is.
- If we have a model $\beta_0 + \beta_1 x_1^\alpha + g(\beta_2, \ldots, \beta_k; x_2, \ldots, x_k) + \varepsilon$ we can use the same idea as above but replace $f(\xi, \beta_0, \beta_1) = \beta_0 + \beta_1 \xi(\alpha)$ with $f(\xi, \boldsymbol{\beta}) = \beta_0 + \beta_1 \xi(\alpha) + g(\beta_2, \ldots, \beta_k; x_2, \ldots, x_k)$.

Example

```
alpha0 = .1

for (j in c(1:10)) {

  df01.model0 <- lm(people_fully_vaccinated_per_hundred~I(
    gdp_per_capita^alpha0), data = df01)

  df01.model1 <- lm(people_fully_vaccinated_per_hundred~I(
    gdp_per_capita^alpha0) + I(gdp_per_capita^alpha0*log(gdp
    _per_capita)), data = df01)

  alpha0 <- alpha0 + df01.model1$coefficients[[3]]/df01.
    model0$coefficients[[2]]
}

cat("alpha: ", alpha0)
```

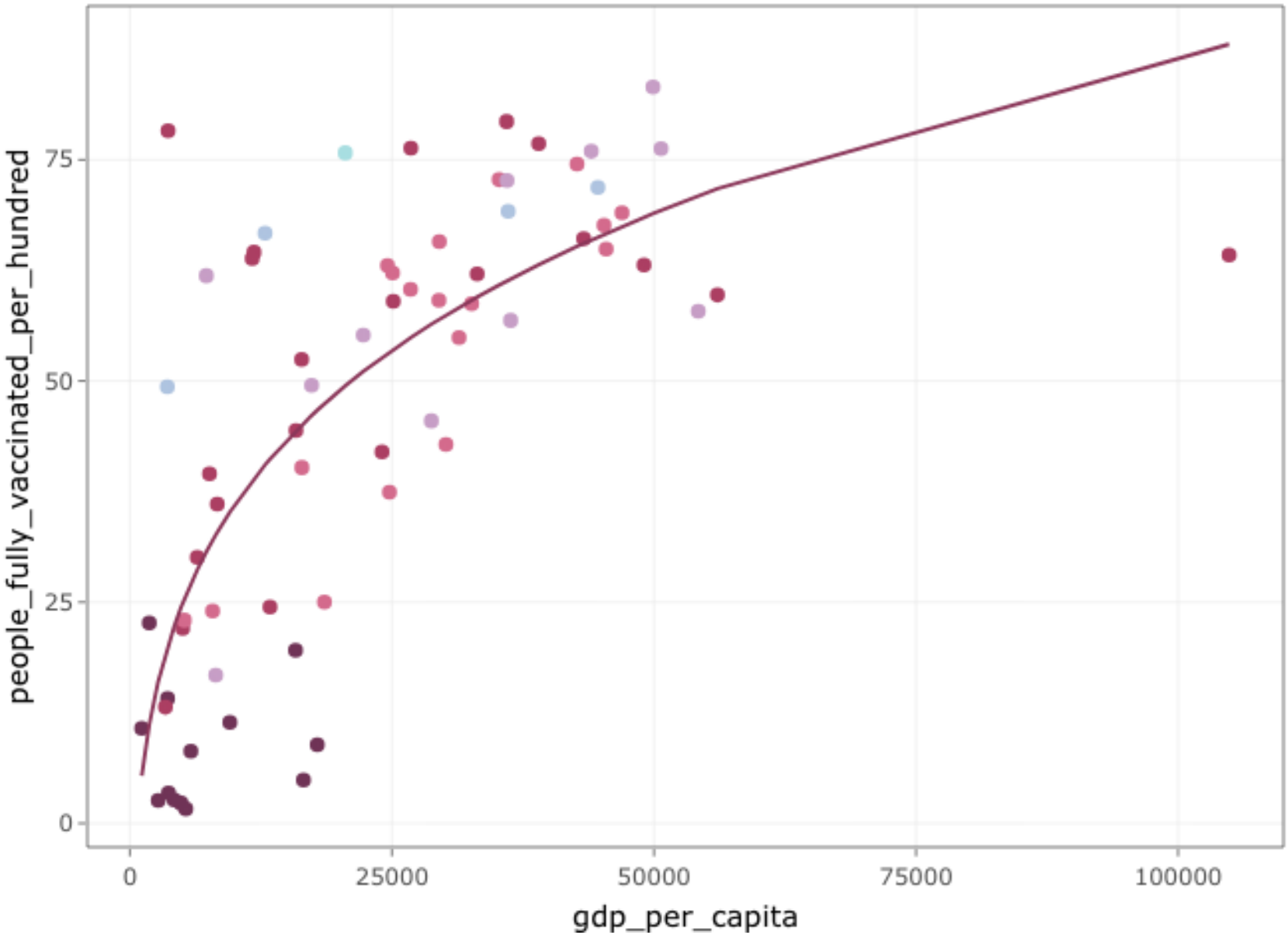
```
alpha:  0.195651
```

Example

```r
df01.model0 <- lm(people_fully_vaccinated_per_hundred~I(gdp_
    per_capita^.2), data = df01)
summary(df01.model0)
```

```
Call:
lm(formula = people_fully_vaccinated_per_hundred ~ I(gdp_per
    _capita^0.2), data = df01)

Residuals:
    Min      1Q  Median      3Q     Max
-40.313 -12.263   1.210   8.928  58.120

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           -50.771     11.601  -4.377 4.39e-05 ***
I(gdp_per_capita^0.2)  13.752      1.609   8.549 2.77e-12 ***
---
Signif.codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.26 on 66 degrees of freedom
Multiple R-squared:  0.5255,^^IAdjusted R-squared:  0.5183
F-statistic: 73.08 on 1 and 66 DF,  p-value: 2.766e-12
```

Example


people_fully_vaccinated_per_hundred
0
25
50
75
0
25000
50000
75000
100000
gdp_per_capita

Example

```
1 out <- boxcoxnc(df01$people_fully_vaccinated_per_hundred,
      df01$gdp_per_capita^.1, method = "mle", lambda = seq
      (-2,2,0.0001), verbose = T, plot = F)
```

```
Box-Cox power transformation
---------------------------------------------------------------
data : df01$people_fully_vaccinated_per_hundred

lambda.hat : 1.1176


Shapiro-Wilk normality test for transformed data (alpha =
    0.05)
---------------------------------------------------------------

statistic  : 0.9096499
p.value    : 0.0001193572

Result     : Transformed data are not normal.
---------------------------------------------------------------
```