SF2930 - Regression analysis

KTH Royal Institute of Technology, Stockholm

Lecture 7 – Methods for detecting influential observations (MPV 6, 9)

March 13, 2022

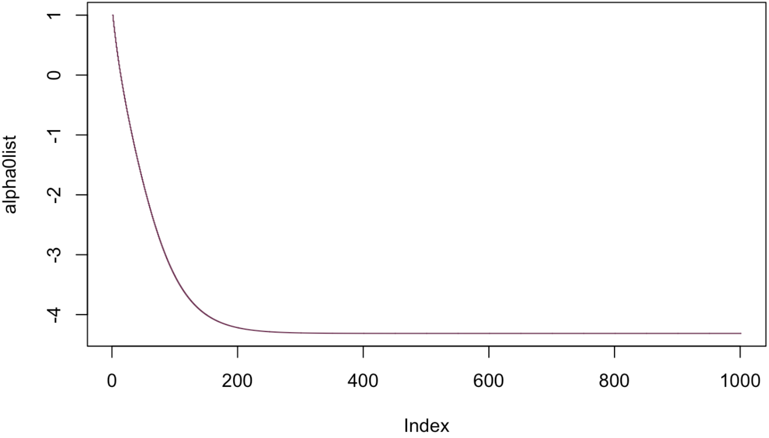
Todays lecture

- More on the "best power" transform
- High leverage points vs. outliers
- Using H to find high leverage points
- Methods for finding influential points: Cook's distance, DFBETAS, and DFFITS
- Multicollinearity, definition, sources, and effects

Finding the "correct power"

```r
firstpower <- 0.1936642

alpha0 <- 1
alpha0list  <-  c(alpha0)
dt <- .01
T <- 1000

for (j in c(1:T)) {
df01.model0 <- lm(people_fully_vaccinated_per_hundred~I(
    diabetes_prevalence^alpha0) + I(gdp_per_capita^
    firstpower), data = df01)
df01.model1 <- lm(people_fully_vaccinated_per_hundred~I(
    diabetes_prevalence^alpha0) + I(diabetes_prevalence^
    alpha0*log(diabetes_prevalence)) + I(gdp_per_capita^.1),
     data = df01)


alpha0 <- alpha0 + dt* df01.model1$coefficients[[3]]/df01.
    model0$coefficients[[2]]
alpha0list <- append(alpha0list,c(alpha0))
}

plot(alpha0list,type = "s", col="#703457")
cat("alpha: ", alpha0)
```

Finding the "correct power"



Finding the "correct power"

```r
summary(lm(people_fully_vaccinated_per_hundred~ I(gdp_per_
    capita^.2) + I(diabetes_prevalence^-4), data = df01) )
```

```
lm(formula = people_fully_vaccinated_per_hundred ~ I(gdp_per
    _capita^0.2) +
    I(diabetes_prevalence^-4), data = df01)

Residuals:
    Min       1Q   Median       3Q      Max
-28.089   -7.267   -1.172    5.255   53.035

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          -57.032     14.407  -3.959 0.000259 ***
I(gdp_per_capita^0.2) 14.692      1.915   7.674 8.98e-10 ***
I(diabetes_prev^-4) 1663.632   1422.056   1.170 0.248076
---
Signif.codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 15.46 on 46 degrees of freedom
Multiple R-squared:  0.5667,^^IAdjusted R-squared:  0.5479
F-statistic: 30.08 on 2 and 46 DF,  p-value: 4.428e-09
```

Finding the "correct power"

```r
library("car")
boxTidwell(people_fully_vaccinated_per_hundred ~ gdp_per_capita + diabetes_prevalence, data=df01, tol=0.00001, verbose=FALSE, max.iter=100)
```

```
                MLE of lambda Score Statistic (z) Pr(>|z|)
gdp_per_capita        0.38862             -2.5506  0.01075 *
diabetes_prevalence   4.00294              1.4445  0.14860
---
Signif.codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

iterations =  15
```

Finding the "correct power"

```r
summary(lm(people_fully_vaccinated_per_hundred~ I(gdp_per_
    capita^.388) + I(diabetes_prevalence^4), data = df01) )
```

```
Call:
lm(formula = people_fully_vaccinated_per_hundred ~ I(gdp_per
    _capita^0.388) + I(diabetes_prevalence^4), data = df01)

Residuals:
    Min      1Q  Median      3Q     Max 
-37.755 -12.768   0.958   8.540  60.352 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -1.288e+01  6.586e+00  -1.956   0.0549 .  
I(gdp_per_c^0.388) 1.277e+00  1.368e-01   9.335 1.48e-13 ***
I(diabetes_pr^4 )  6.400e-05  2.568e-05   2.492   0.0153 *  
---
Signif.codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.36 on 64 degrees of freedom
Multiple R-squared: 0.5826, Adjusted R-squared:  0.5695 
F-statistic: 44.66 on 2 and 64 DF,  p-value: 7.231e-13
```

High leverage points vs. outliers

Influential points

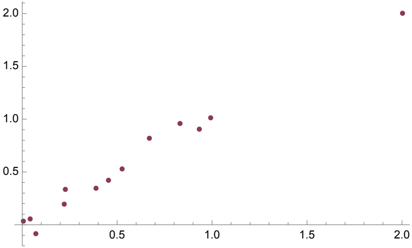
Data point which affects the choice of model a lot.

High leverage points

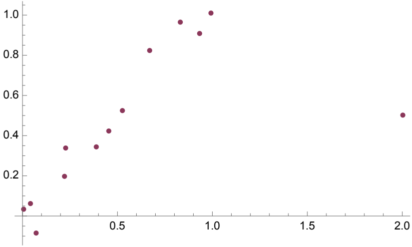
Data points which exerts a lot of "pull" on the model.

Outliers

Data points whose response does not follow the general trend of the data. Might be due to errors, fat tailed error distributions, etc. Often we want to remove them, but must be transparent that this was done.



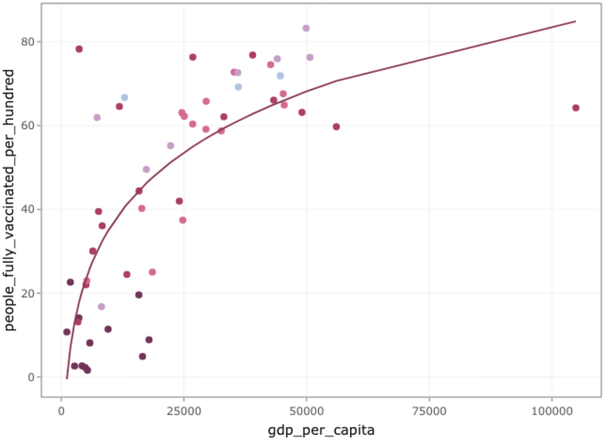
A high leverage point



An outlier?

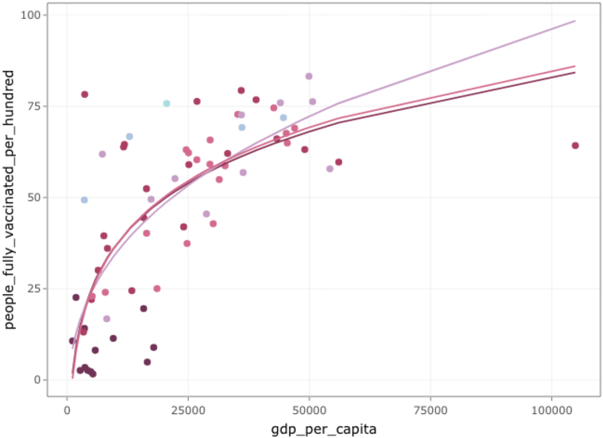
Example

```r
scaledgdp <- (df01$gdp_per_capita)^0.1
df01$scaled_gdp_per_capita <- scaledgdp

df01.model2 <- lm(people_fully_vaccinated_per_hundred~scaled
    _gdp_per_capita, data = df01)
```



Example

```
scaledgdp <- (df01$gdp_per_capita)^0.1
df01$scaled_gdp_per_capita <- scaledgdp

df01.model2 <- lm(people_fully_vaccinated_per_hundred~scaled
    _gdp_per_capita, data = df01)
```

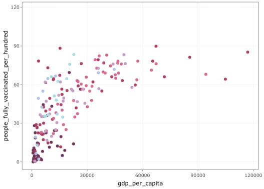


Example

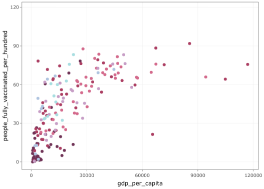
```r
df <- read.csv("/Users/malin/Dropbox/Jobb/Teaching/KTH - SF2930/data.csv", header = TRUE)
```

```r
df0 <- df0 %>% group_by(location) %>% slice(n()) %>% ungroup

df0[df0$location=="Norway",8] <- 71.2
df0[df0$location=="Qatar",8] <- 85.2
...
```

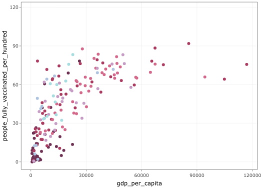
```r
df00 <- df %>% group_by(location) %>% slice_max(people_fully_vaccinated_per_hundred)  %>% slice(n()) %>% ungroup
```



df0



df00



df00 corrected

Using H to find high leverage points

Recall that $H = X(X^T X)^{-1} X^T$.

The hat matrix and variance

$\mathrm{Var}(\mathbf{e}) = \sigma^2(I - H)$ and $\mathrm{Var}(\hat{\mathbf{y}}) = \sigma^2 H$.

Leverage

Since $\hat{\mathbf{y}} = H\mathbf{y}$, h_{ii} is often interpreted as the amount of leverage exerted by y_i on $\hat{y}_i$.

The regressor variable hull

When the columns of X are normalized, $h_{ii} = \mathbf{x}_i^T (X^T X)^{-1} \mathbf{x}$ is often used a standardized measure of the distance between $\mathbf{x}_i$ and the center of the space of $\mathbf{x}$-values.

Properties of the diagonal of H

- $-1/n < h_{ii} \leq 1$ ($0 < h_{ii} \leq 1$ if there is no intercept term)
- $\sum h_{ii} = rank H = rank X = k + 1$, and hence $\bar{h} = (k+1)/n$. (k instead of $k + 1$ if there is no intercept)

High leverage points

If $|\bar{h}| < 1$, we say that $(\mathbf{x}_i, y_i)$ is a *high leverage point* if $h_{ii} > 2\bar{h}$.

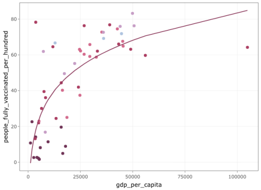
Influential points

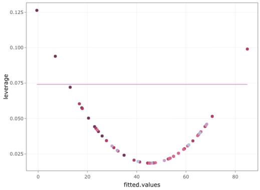
Observations with h_{ii} high are only influential if they also have high residuals.

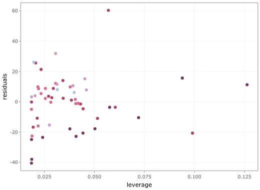
Example

```r
df01.model2.extra <- df01[,c("continent","location","gdp_per_capita")]
df01.model2.extra$leverage <- hatvalues(df01.model2)
df01.model2.extra$residualsstudentized <- rstudent(df01.model2)

df01.model2.extra[which(df01.model2.extra[,"leverage"]>2*hatmean),]
```

continent	location	gdp_per_capita	leverage	residualsstudentized
Asia	Macao	104861.851	0.09901327	-1.2362750
Africa	Mozambique	1136.103	0.12631339	0.6759785
Africa	Rwanda	1854.211	0.09393737	0.9283018







Cook's distance

A measure which consider both the location of a point and its effect.

Cook's distance

$$D_i := \frac{(\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}})^T X^T X (\hat{\boldsymbol{\beta}}_{(i)} - \hat{\boldsymbol{\beta}})}{(k+1)MS_{Res}} = \frac{r_i^2}{k+1} \cdot \frac{h_{ii}}{1-h_{ii}} = \frac{(\hat{\mathbf{y}}_{(i)} - \hat{\mathbf{y}})^T (\hat{\mathbf{y}}_{(i)} - \hat{\mathbf{y}})}{(k+1)MS_{Res}}$$

(Here r_i is the internally studentized residual at i.)

Cutoff motivation

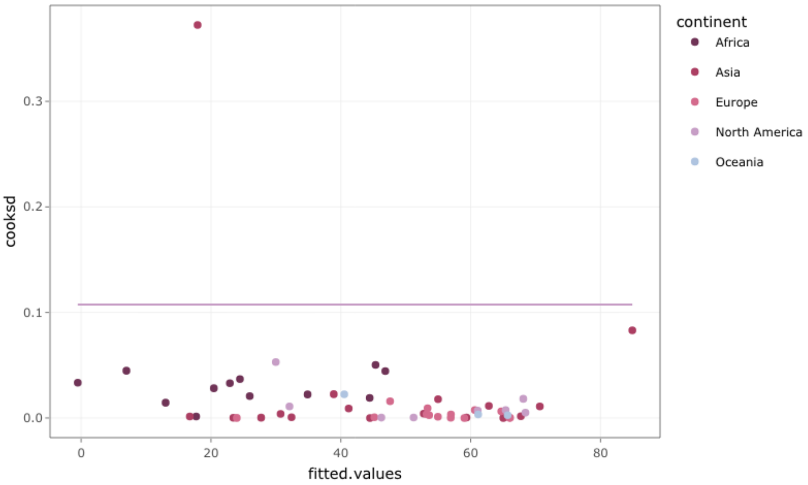
$D_i \sim F_{k+1,n-k-1}$. Let α be such that $F_{\alpha,k+1,n-k-1} = D_i$. Then, heuristically, deleting point i would move $\hat{\boldsymbol{\beta}}$ to $\hat{\boldsymbol{\beta}}_{(i)}$, which lies on the boundary of a $100\alpha\%$-confidence region for $\boldsymbol{\beta}$.

Influential points

We say that $(\mathbf{x}_i, y_i)$ is influential if $D_i > F_{\alpha_0,k+1,n-k-1}$.

Example

```r
cooks.distance(df01.model2)
```



DFBETAS

A measure of how much the ith observation affects $\hat{\beta}_j$.

DFBETAS

$$DFBETAS_{ij} := \frac{\hat{\beta}_j - \hat{\beta}_{j(i)}}{\sqrt{S_{(i)}^2 (X^T X)^{-1}(j,j)}} = \frac{R_{ji}}{\sqrt{R_{j\cdot}^T R_{j\cdot}}} \cdot \frac{t_i}{\sqrt{1 - h_{ii}}}$$

where $R = (X^T X)^{-1} X^T$ and t_i is the externally studentized residual at i. $R_{j\cdot}$ can be thought of as a measure of the leverage the points in the sample has on $\hat{\beta}_i$.

$DFBETAS_{ij}$ measures how much $\hat{\beta}_j$ changes if the ith observation is deleted, rescaled by the standard deviation of $\hat{\beta}_j$. In other words, it measures the effect observation i has on $\hat{\beta}_j$.

Suggested cutoff

$|DFBETAS_{ij}| > 2/\sqrt{n}$.

Example

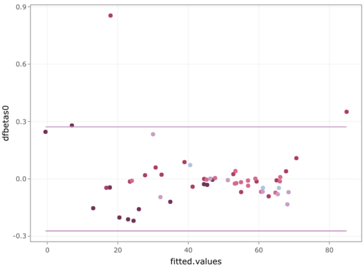
```r
df01.model2.extra$dfbetas0 <- dfbetas(df01.model2)[,1]
df01.model2.extra$dfbetas1 <- dfbetas(df01.model2)[,2]

threshold <- 2/sqrt(nrow(df01.model2.extra))
df01.model2.extra[which(df01.model2.extra[,"dfbetas0"]>
    threshold | df01.model2.extra[,"dfbetas0"]< (-threshold)
    ),]
df01.model2.extra[which(df01.model2.extra[,"dfbetas1"]>
    threshold | df01.model2.extra[,"dfbetas1"]< (-threshold)
    ),]
```

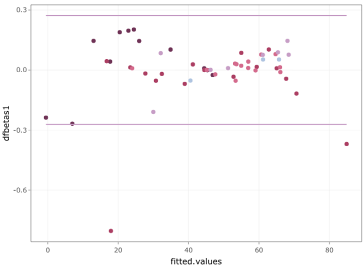
continent	location	dfbetas0	dfbeta1	residualsstudentized
Asia	Cambodia	0.8540861	-0.6714375	3.980970
Asia	Macao	0.3506097	-0.5641862	-1.236275
Africa	Rwanda	0.2794231	0.2085492	-0.2678244

continent	location	dfbetas0	dfbeta1	residualsstudentized
Asia	Cambodia	0.8540861	-0.8038152	3.980970
Asia	Macao	0.3506097	-0.3695215	-1.236275

Example

```r
df01.model2.extra$dfbetas0 <- dfbetas(df01.model2)[,1]
df01.model2.extra$dfbetas1 <- dfbetas(df01.model2)[,2]

threshold <- 2/sqrt(nrow(df01.model2.extra))
df01.model2.extra[which(df01.model2.extra[,"dfbetas0"]> threshold | df01.model2.extra[,"dfbetas0"]< (-threshold)),]
df01.model2.extra[which(df01.model2.extra[,"dfbetas1"]> threshold | df01.model2.extra[,"dfbetas1"]< (-threshold)),]
```





DFFITS

A measure of how much the ith observation affects $\hat{y}_i$.

DFFITS

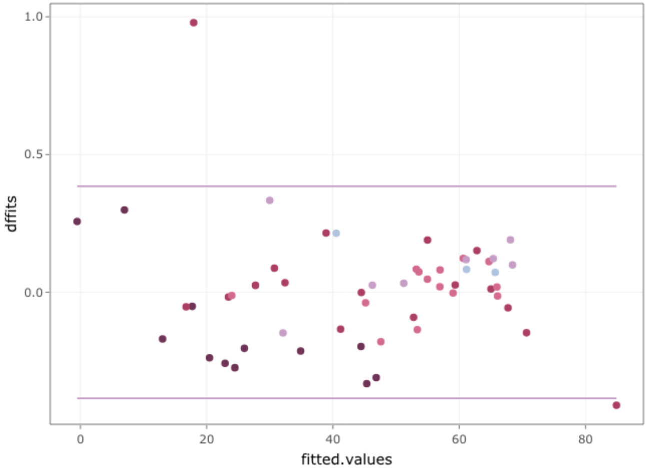
$$DFFITS_i := \frac{\hat{y}_i - \hat{y}_{(i)}}{\sqrt{S^2_{(i)} h_{ii}}} = \left(\frac{h_{ii}}{1 - h_{ii}}\right)^{1/2} t_i.$$

Suggested cutoff
$|DFFITS_i| > 2\sqrt{(k+1)/n}$.

Example

```r
dffits(df01.model2)
```

continent	location	dffits	residualsstudentized
Asia	Cambodia	0.9784329	3.980970
Asia	Macao	-0.4098289	-1.236275



How to use these statistics?

- Cut-offs should be set so that we get a realistic number of influential points.
- We should only remove data-points if we are sure they are outliers, which needs investigation. No automatic test can show if a point is an outlier. Rather, they detect points that could be outliers. We always need to investigate such points further (manually) by e.g. looking more carefully at the data, before w e classify them as outliers and remove them from our data set.
- These methods are particularly useful for very large datasets, where we cannot "see" the outliers directly in, e.g., plots.

Multicollinearity

A dataset is said to exhibit multicollinearity if there is a new-linear relationship between the regressors.

Linear dependence

There is a linear dependence in X if there is non-zero $t_1, t_2, \ldots, t_n$ such that $t_0 X_{1\cdot} + t_1 X_{1\cdot} + \ldots + t_k X_{k\cdot} = 0$, i.e. such that $X\mathbf{t} = 0$.

Near linear dependence

There is multicollinearity in X if there is $\mathbf{t} \neq 0$ such that $X\mathbf{t} \approx 0$.

Effects of multicollinearity

What happens if there is a linear dependence?

Recall that the LSE (and MLE) of $\hat{\boldsymbol{\beta}}$ minimizes $\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2$, and that the minimum is a solution to $X^T X \hat{\boldsymbol{\beta}} = X^T \mathbf{y}$.

- If there is a linear dependence in X, then X does not have full rank, and hence $X^T X$ is not invertible.
- If $X^T X$ is not invertible, then $X^T X \hat{\boldsymbol{\beta}} = X^T \mathbf{y}$ has infinitely many solutions. In other words, there exists infinitely many equally good choices of regression coefficients $\beta_0, \beta_1, \ldots$

Effects of multicollinearity

What happens if there is almost a linear dependence?

- If there is almost a linear dependence in X, then $\det X^T X$ is "small" but non-zero. Hence $(X^T X)^{-1}$ exist, but has some very large entries.
- The equation $X^T X\hat{\boldsymbol{\beta}} = X^T \mathbf{y}$ will have a unique solution $\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{y}$, but this solution will be very sensitive to small changes in X, such as measurement errors or calculation errors made by the computer. We will see this by noting that the confidence intervals for β_j, given by

$$\beta_j = \hat{\beta}_j \pm t_{\alpha/2,n-k-1} \sqrt{\hat{\sigma}^2 (X^T X)^{-1}(j,j)}$$

  become very large.

Effects of multicollinearity

What happens if there is almost a linear dependence?

$$\mathbb{E}\big[\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}\|_2^2\big]=\mathbb{E}[(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})^T(\hat{\boldsymbol{\beta}}-\boldsymbol{\beta})]=\sum\mathbb{E}[(\hat{\beta}_j-\beta_j)^2]=\sum\operatorname{Var}\hat{\beta}_j$$
$$=\sum\sigma^2(X^TX)^{-1}(j,j)=\sigma^2\operatorname{tr}(X^TX)^{-1}=\sigma^2\sum\frac{1}{\lambda_j}$$

or equivalently,

$$\mathbb{E}[\|\hat{\boldsymbol{\beta}}\|_2^2]=\|\boldsymbol{\beta}\|_2^2+\sigma^2\operatorname{tr}(X^TX)^{-1}=\|\boldsymbol{\beta}\|_2^2+\sigma^2\sum\frac{1}{\lambda_j}$$

If X^TX is almost not invertible, then at least one of the eigenvalues of X^TX is going to be very close to zero, and hence at least one of the regression coefficients is likely to be far from the correct value, and the vector of regression coefficients will in general be too long.

Example

```
df01 <- df01[!is.na(df01[,"diabetes_prevalence"]),]

df01$scaled_gdp_per_capita <- df01$gdp_per_capita^.1
df01$scaled_diabetes_prevalence <- df01$diabetes_prevalence^1.25

X <- data.matrix(df01[,c("scaled_gdp_per_capita", "scaled_diabetes_prevalence")])
X <- cbind(ones=1,X);
XtX <- t(X)%*%X
eigen(XtX)$values
```

```
[1] 1.526829e+04 1.268325e+02 4.014914e-01
```

Example

```r
df01 <- df01[!is.na(df01[,"diabetes_prevalence"]),]

df01$scaled_gdp_per_capita <- df01$gdp_per_capita^.1-mean(df01$gdp_per_capita^.1)
df01$scaled_gdp_per_capita <- df01$scaled_gdp_per_capita/sqrt(sum((df01$scaled_gdp_per_capita)^2))

df01$scaled_diabetes_prevalence <- df01$diabetes_prevalence^1.25-mean(df01$diabetes_prevalence^1.25)
df01$scaled_diabetes_prevalence <- df01$scaled_diabetes_prevalence/sqrt(sum((df01$scaled_diabetes_prevalence)^2))

X <- data.matrix(df01[,c("scaled_gdp_per_capita", "scaled_diabetes_prevalence")])

XtX <- t(X)%*%X

eigen(XtX)$values
```

```
[1] 1.2674516 0.7325484
```

Example

```r
df04 <- df00[!is.na(df00[,"total_cases_per_million"]) & !is.na(df00[,"total_deaths_per_million"]) & !is.na(df00[,"median_age"]) & !is.na(df00[,"gdp_per_capita"]) & !is.na(df00[,"hospital_beds_per_thousand"]) & !is.na(df00[,"people_fully_vaccinated_per_hundred"]) & !is.na(df00[,"population_density"]) & !is.na(df00[,"male_smokers"]) & !is.na(df00[,"diabetes_prevalence"]) & !is.na(df00[,"life_expectancy"]),]


X <- data.matrix(df04[,c("total_cases_per_million","total_deaths_per_million","median_age","gdp_per_capita" , "hospital_beds_per_thousand","people_fully_vaccinated_per_hundred", "population_density","male_smokers","life_expectancy" ,"diabetes_prevalence")])
X <- scale(X)/sqrt(nrow(X)-1)

XtX <-  t(X)%*%X

det(XtX)
eigen(XtX)$values
```

```
[1] 4.583689e-05
[1] 4.64521489 2.23059956 1.24750008 0.78877620 0.40032595
    0.28476294 0.19479126 0.10200313 0.08174614 0.02427984
```

Sources of multicollinearity

Constraints on the model or population

Some regressors are naturally related. In the covid data set, we for example expects some things to be related, such as the number of covid cases, the number of covid deaths and the median age of the population for example.

Model specification

If the range of x_{ij} is small, then the two vectors $(x_{ij})_j$ and $(x_{ij}^2)_j$ will be nearly dependent.

An overdefined model

In some cases, it is easy to collect a lot of different regressors, but hard to get a large sample (e.g. medical data). In the extreme case, when $n < k + 1$, we will always get linear dependence.

The data collection method

This occur when we for some reason only sample points in a subset which cause multicollinearity.