Practical Course in Stochastics II

Prof. Dr.Tatyana Krivobokova

Dr.Marco Singer

Wallace Agyei

June 4, 2020

General remarks

- Each part of the exercises consists in sub-questions. We answered all the questions in the order as as it was asked to have more clarity and better explanations.
- We have directly answered all the questions from the exercises but some general background have been omitted from the this report.
- All the figures and plots completely come from the codes, which are well explained in $.R$ files, thus the focus on report was on the description of the results. Although, we describe the some codes detail in this report.
- In order to make the figures and plots reproducible we set the seed to 42 for all the problems,unless the exercise states to change the seed.
- We comment the lines in the codes that were used to save the plots in a specific path of my computer in order to have no problems when running in another computer.

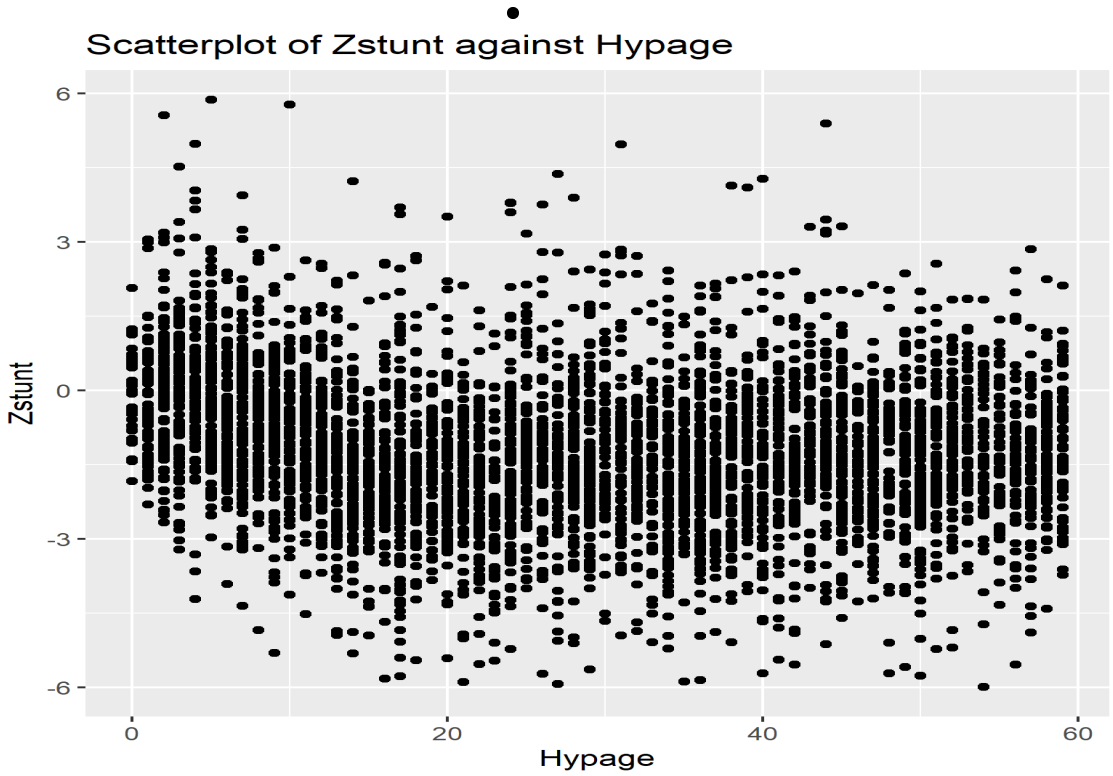
1 Tidyverse

(a) The first task for this exercise essentially involves reading data childrenfinal.dta given in the STATA format. The data was imported in R using an appropriate function from tidyverse.The function read.dta was used to read a STATA format table and convert it into a dataframe which was assign *childrenData*.

- The data has 4686 observations and 177 variables.The variables with starting-names "s", "v" and "m" with some attached digits were deleted using *pipes* and *select* functions from the R package.
- In addition, where necessary, the variable types are converted as suitable. For instance, the variables ruralfacto and female have been change to factor variables.
- Lastly, the dataframe was converted to a tibble frame called *ChildrenData*1.

(b) In the second task we create smaller tibble of the main data that contains variables hypage, ruralfacto, female, zstunt,zweight, zwast, adm2 called *ChildrenData*2 using *pipes* and *select* functions.The variable zstunt is the so-called Z-score for stunting and is defined as the height of a child standardised with the median and standard deviation of heights of children at the same age from a healthy population.

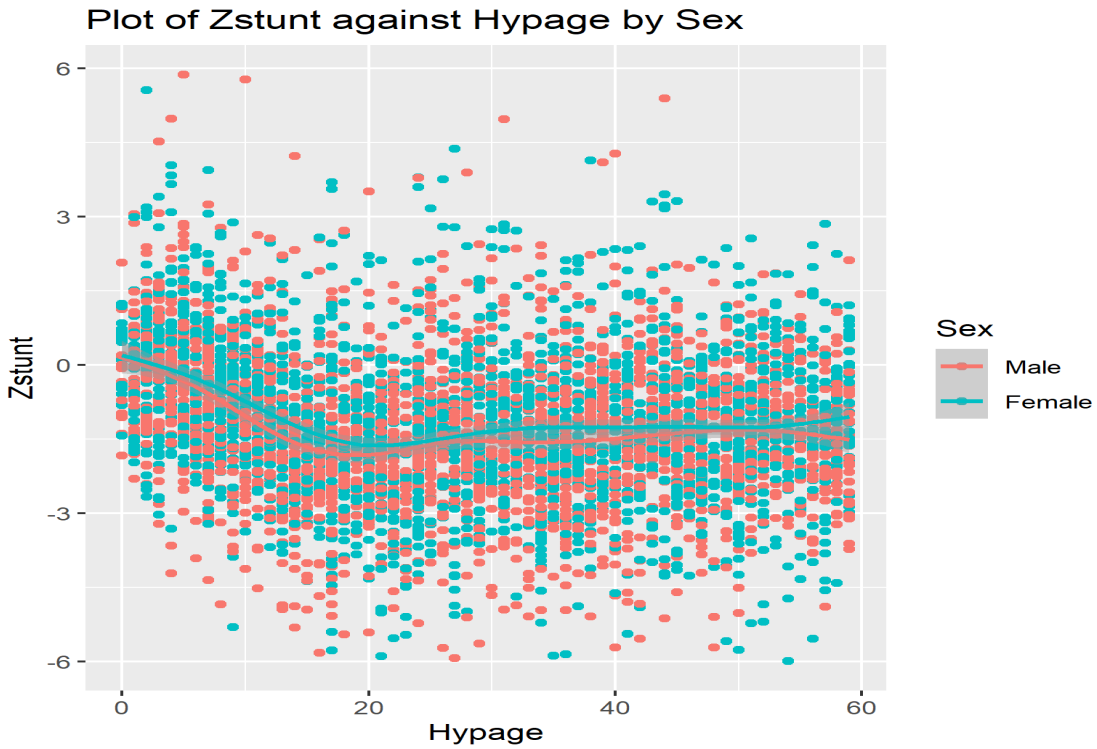
- The figure below is the scatter plot of the variable Zstunt against Hypage where children with Z-score less than -2 are defined to be stunted.



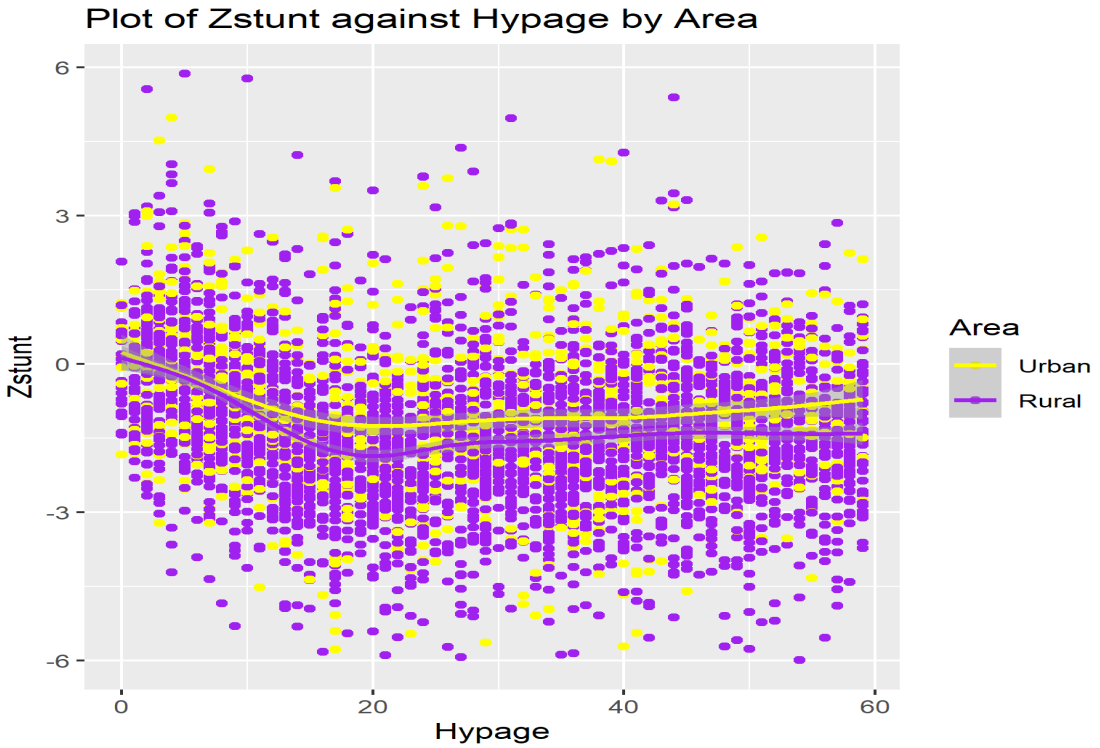
In the plot we can see that since the child is born there is a decreasing of the z score variable that reachs the minimum bound when the child is around 18 months. After that, we can see an slightly increasing until 25 months and then the curve maintains flat. We can infer that the risk of the child is given until 20 months.

1. We now make smooth plots of zstunt against age for females and males on one plot, with a suitable legend using different colors for males and females.

   - Showing the plot.



   In the plot we can see that the z score curve of females and males are almost parallel curves and both follow similar behavior than the previous plot. Since the male curve is under the female curve we can say that male children have more risk than females.



]

Similar to the plot in plot 2, we can see that the z score curve of children from Urban and Rural are almost parallel curves and both follow similar behavior. Since the rural curve is under the urban curve we can say that rural children have more risk than urban.

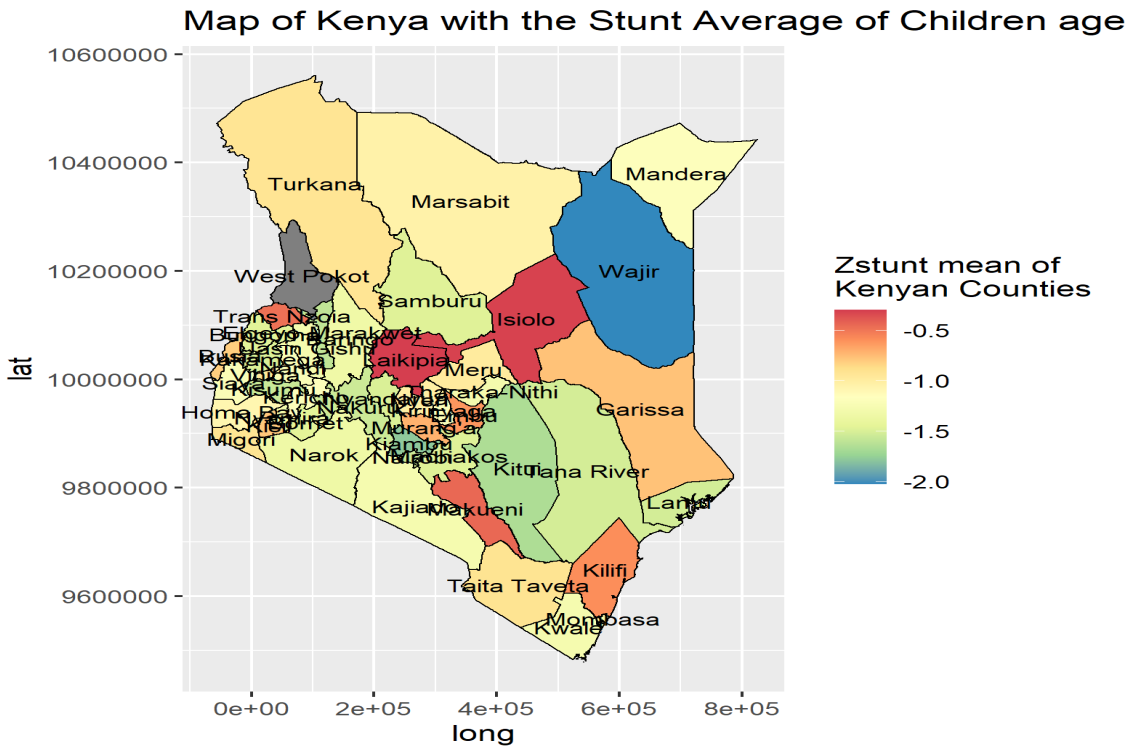
(c) The next task of this exercise was to plot the map of Kenya with all the counties based on the mean of the Zstunt of each county. In the original data, the county named Isolo was missing. We have factored this into the data. We would like to observe, graphically, the county with stunted children(children with Zscore less than -2). The map, as drawn on R, is shown in fi

g.4:

(c) In addition the table below present 3 counties with less risk (top 3) and 3 counties with the highest (bottom 3). We can see that the county which in average is stunted is West Pokot.

County List			
County Name	Longitud center	Lattitud center	zstunt
LAMU	729192.144229795	9770610.67554005	-0.299
KAJIADO	269565.423314458	9792732.95575617	-0.311
MANDERA	693905.334241803	10334715.9287016	-0.445
KWALE	547286.76709441	9519240.85546736	-1.66
KILIFI	602753.762807076	9623609.13786212	-1.78
WEST POKOT	103214.242423623	10172910.9801468	-2.02

(d) The last part of this exercise, was to write the tibble from (b) into a text file. This file will be used in Exercise 5.

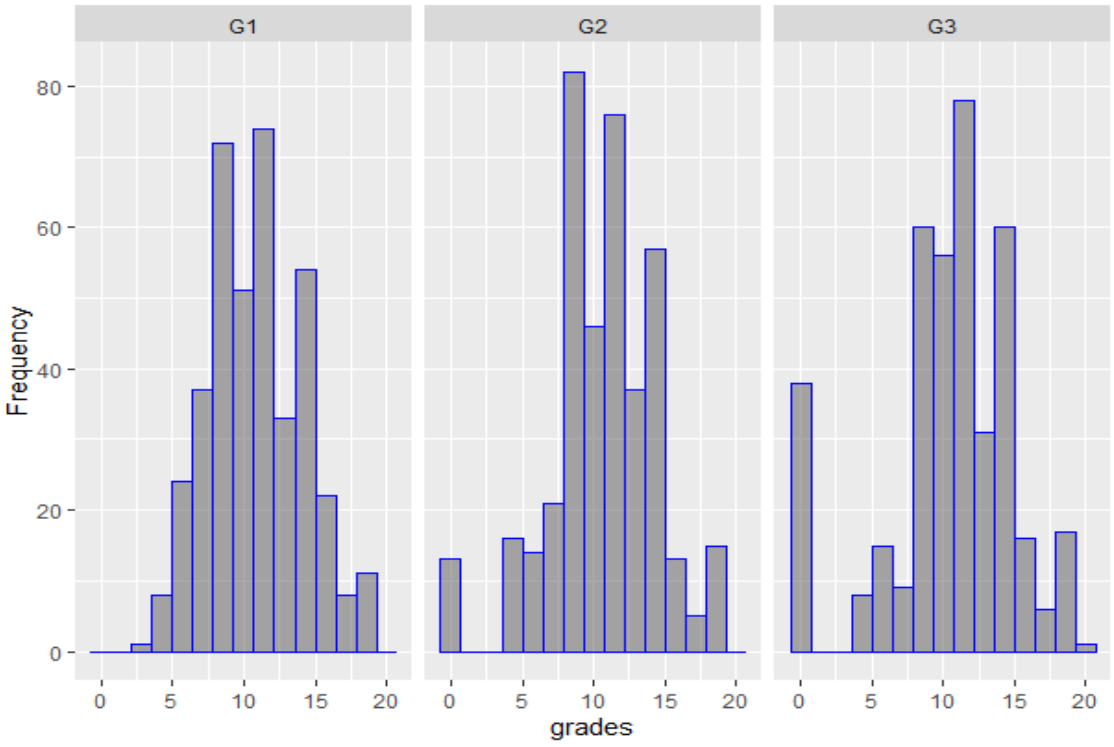


- For this purpose we use the R-function *write.table*

2 Generalised linear models

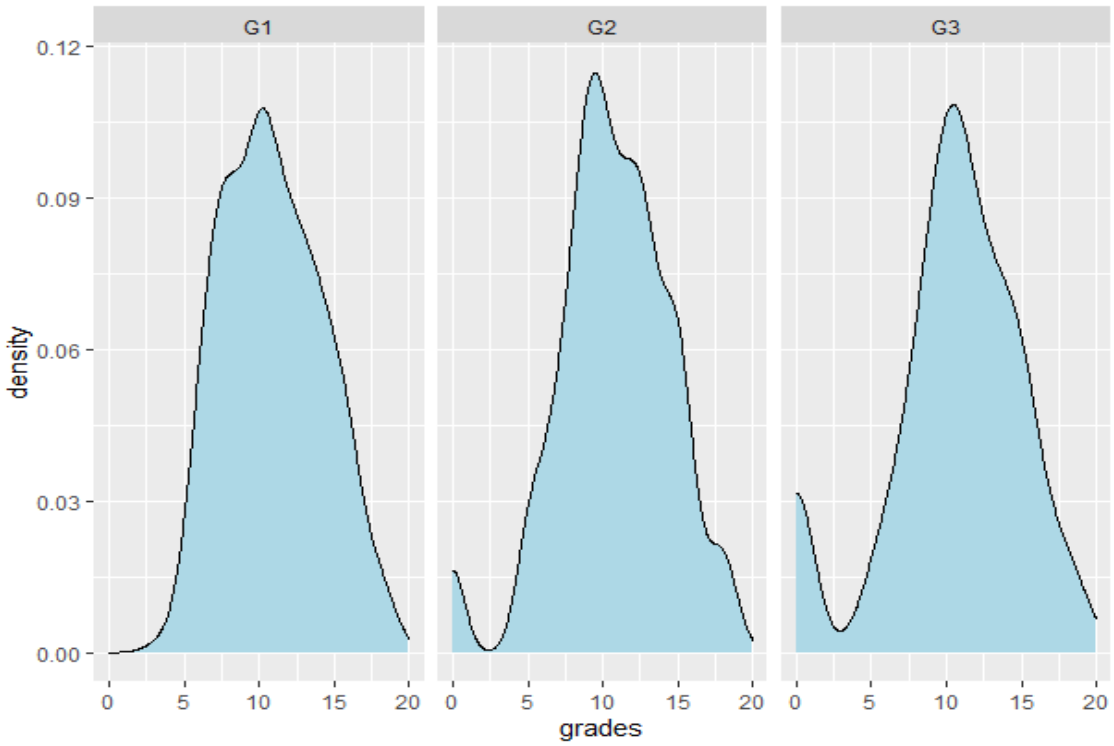
(a) In this exercise we examine the normality or otherwise of these variables. We explored the tools of histogram and empirical densities to ascertain behaviour of the variables grades G1,G2 and G3.

- Histogram:



We can see that we have grades zero in all G1,G2 and G3. We should remove this values and take only grades greater than zero since we might have observations that were not measure under the "preparation" factor. Now, it is worth to mention that we are handling a discrete variable so we have to fit it to a discrete distributions. Although, the plot specifically for G1 shows an approximate normal density.We present normal fitting plots for all grades to verify if they follow normal distribution.

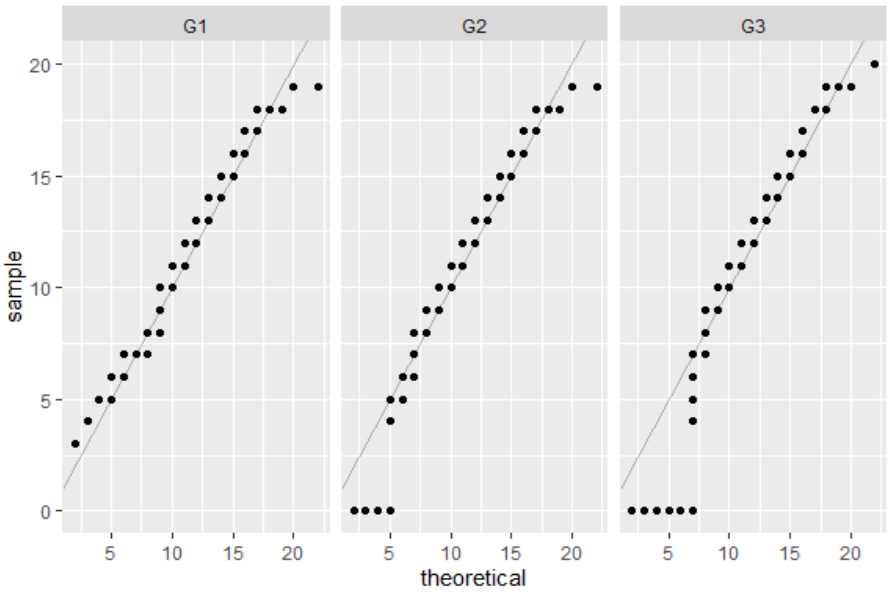
- Density plots:



It seems that the sample have a right tail where a Poisson would be more suitable. In the next question we will find this out by using the goodness of fit.

1. For each of G1, G2, G3 we check to follow a Poisson distribution or not using Q-Q plot.

   - Q-Q plots:



   - From the Q-Q plots above it is obvious the G1,G2 and G3 are not poisson distributed

- The next task was to check whether the three variables are over-dispersed or not. We calculated the mean-variance ratios and obtain 1.01,1.32, and 2.02 for G1,G2 and G3 respectively. From the results we can argue that the three variables are over-dispersed since their mean-variance ratios are greater than 1 which implies the variances are greater than the means.

- The second task for this exercise was to fit a generalized linear model to explain G1 with all the explanatory variables.

- We create a variable model1 using the *glm* function with poisson as the link.

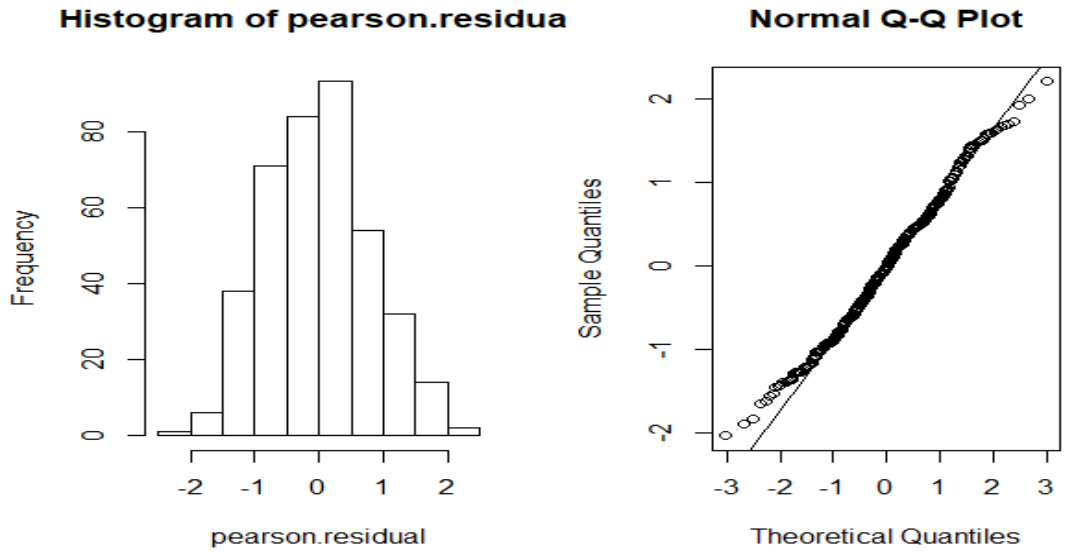
- The summary of fitted model is presented in table below.The table contains the variables and their *beta* coefficients according to the glm model . If the p-value has * it means that the variable is significant.

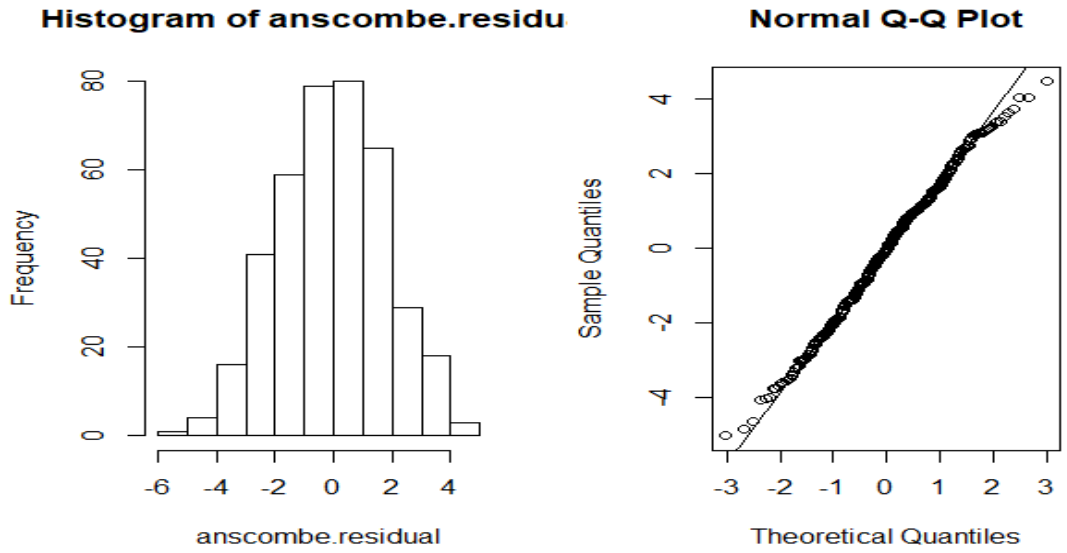
Model 1 coefficients				
beta	Estimate	Std. Error	z value	p-value
(Intercept)	2.418	0.332	7.289	0***
schoolMS	0.007	0.058	0.122	0.903
sexM	0.078	0.036	2.147	0.032*
age	-0.006	0.016	-0.405	0.685
addressU	0.013	0.044	0.294	0.769
famsizeLE3	0.037	0.036	1.029	0.303
PstatusT	0.016	0.053	0.302	0.762
Medu	0.011	0.024	0.448	0.654
Fedu	0.015	0.02	0.732	0.464
Mjobhealth	0.079	0.081	0.966	0.334
Mjobother	-0.077	0.054	-1.417	0.157
Mjobservices	0.044	0.059	0.734	0.463
Mjobteacher	-0.084	0.077	-1.096	0.273
Fjobhealth	-0.042	0.104	-0.407	0.684
Fjobother	-0.1	0.074	-1.355	0.175
Fjobservices	-0.087	0.076	-1.14	0.254
Fjobteacher	0.096	0.091	1.052	0.293
reasonhome	0.016	0.041	0.386	0.7
reasonother	-0.017	0.061	-0.285	0.775
reasonreputation	0.039	0.042	0.914	0.361
guardianmother	-0.003	0.04	-0.067	0.946
guardianother	0.091	0.074	1.228	0.219
traveltime	-0.003	0.026	-0.132	0.895
studytime	0.054	0.021	2.585	0.01**
failures	-0.148	0.028	-5.245	0***
schoolsupyes	-0.212	0.053	-4.027	0***
famsupyes	-0.093	0.035	-2.65	0.008**
paidyes	-0.009	0.035	-0.248	0.804
activitiesyes	-0.007	0.033	-0.218	0.828
nurseryyes	0.005	0.041	0.11	0.913
higheryes	0.123	0.088	1.404	0.16
internetyes	0.02	0.047	0.417	0.676
romanticyes	-0.019	0.035	-0.531	0.596
famrel	0.002	0.018	0.11	0.913
freetime	0.023	0.017	1.326	0.185
goout	-0.037	0.017	-2.239	0.025*
Dalc	-0.002	0.025	-0.082	0.935
Walc	-0.005	0.019	-0.249	0.804
health	-0.015	0.012	-1.305	0.192
absences	0.002	0.002	0.696	0.487

The only significant variables from the previous table which are sex, studytime, failures, schoolsupyes, famsupyes and goout.

- Goodness of fit, the model is significant with p <<0.5 we use the *pchisq* function in R to compare deviance and residuals.

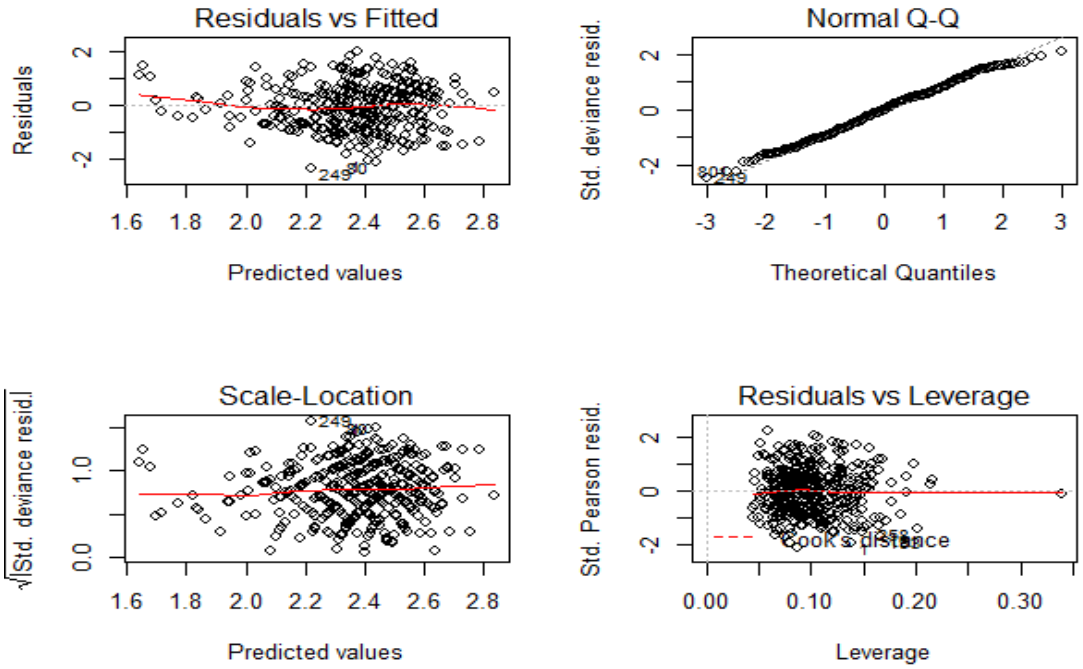
- Residual plots:





We can see in the plots above that both residuals seems to follow a normal distribution but with a slight right-tail. Additionally, in the Q-Q plots above we can also see that in the right side of the line there are points that move away from it.

- We peruse residual analysis of the fitted model1.



- From the above plots we can observe that the residuals appear to behave randomly and in the Q-Q plot all the points lie on line.

- In this task, we create a variable model2 using *glm* function applying only the covariates sex, Fedu, studytime, failures, schoolsup, famsup, goout and with poisson as link as in model1.

- The summary of fitted model2 is presented in table below.

Model 2 coefficients				
beta	Estimate	Std. Error	z value	p-value
(Intercept)	2.84	0.07	30.955	0***
sexM	0.065	0.032	2.034	0.042*
Fedu	0.042	0.014	2.887	0.003**
studytime	0.058	0.019	3.057	0.002**
failures	-0.138	0.024	-5.56	0***
schoolsupyes	-0.198	0.049	-3.983	0***
famsupyes	-0.073	0.032	-2.62	0.023*
goout	-0.035	0.014	-2.506	0.012*

- From the model coefficients it is only the variable Fedu which is not significant.

- For Goodness of fit, the model2 is also significant with p <<0.5 we use the *pchisq* function to compare deviance and residuals.

- Performing analysis of deviance on the two models, we observe a p-value of 0.1858, we can therefore argue that the two models are not significantly different,however model2 is more preferred than model1 because it has less covariates.

- The next task was to create a variable model3 using *glm* function applying only the covariates sex, Fedu, studytime, failures, schoolsup, famsup, Walc and the link as poisson.

- The summary of fitted model3 is presented in the table below.

Model 3 coefficients				
beta	Estimate	Std. Error	z value	p-value
(Intercept)	2.31	0.07	32.083	0***
sexM	0.075	0.032	2.293	0.021*
Fedu	0.040	0.014	2.752	0.005**
studytime	0.052	0.019	2.703	0.006**
failures	-0.141	0.024	-5.67	0***
schoolsupyes	-0.201	0.049	-4.034	0***
famsupyes	-0.074	0.032	-2.29	0.021*
Walc	-0.026	0.012	-2.052	0.04*

- From the table above we see that all model coefficients significant.

- Since model 2 and 3 are not nested we use the AIC to determine which is better for model 2 AIC is 1975.1 and for model 3 AIC is 1977.2. Since both models have the same number of significant variables and the AIC is slighlty lower in the model 2 so we can argue that model 2 delivers better fit than model 3.

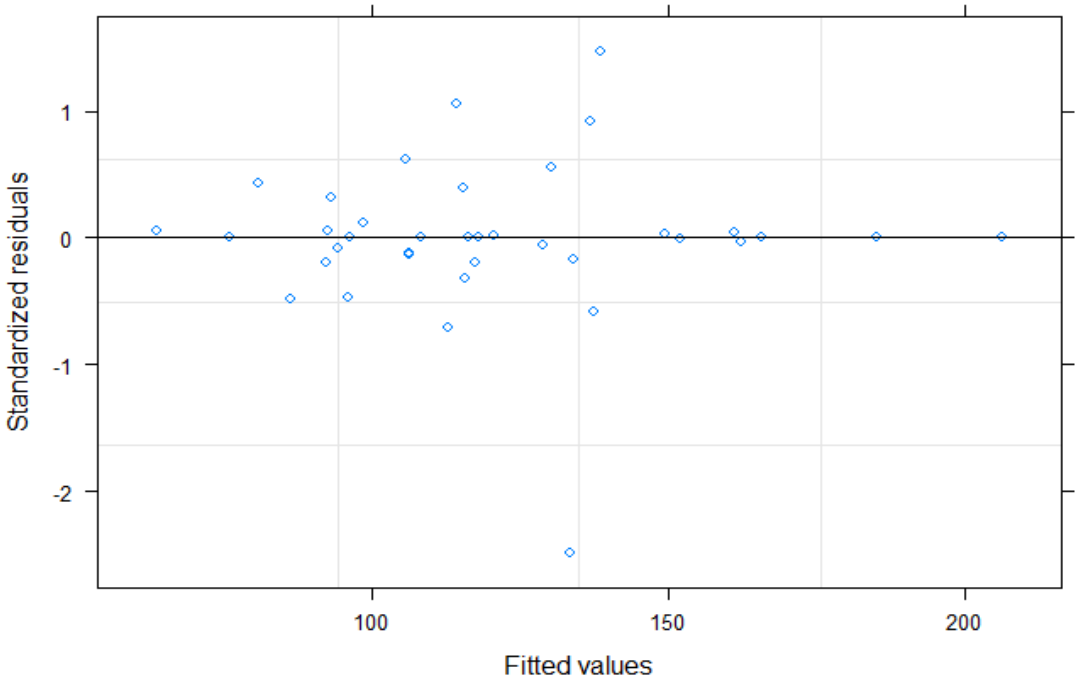
3 Mixed effects models and small area estimation

- In this exercise we use *landsat* dataset.
- The dataset is a survey and satellite data measuring the area for corn and soybeans fields in North-Central Iowa from 1978.
- We are interested in obtaining reliable estimates for the total size of corn and soy production for each of the 12 counties in the data set,
- We choose our parameters HA, as the dependent variable, and Pixels, the independent variable. This is so as the Pixels is a microcosm of the entire produce. It is a unit of the entire crop in the 'farmland'. HA is the whole area for the crops we wish to estimate.
- We fit the linear model for corn (*lm.corn*) and soy (*lm.soy*). The covariates for both of them are the variables *PixelsCorn* and *PixelsSoybeans* since these are the data measured by satelites.

  Model for the Hectareas of Corn:

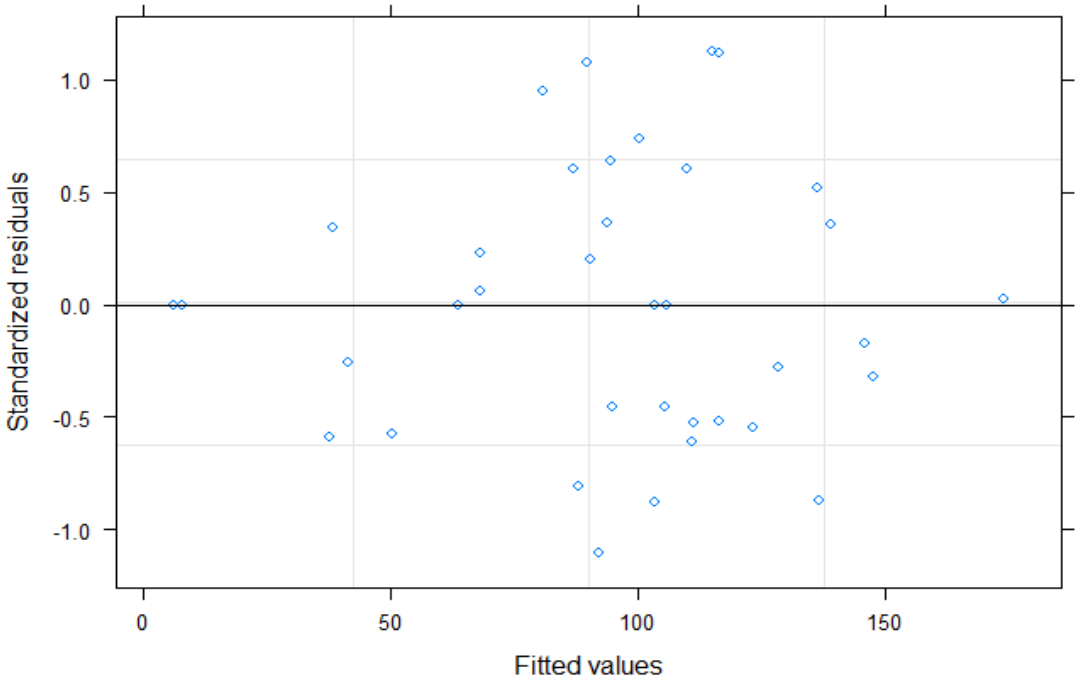
```
> lm.corn
Call:
  Model: HACorn ~ PixelsCorn | CountyName
   Data: corn_group

Coefficients:
             (Intercept) PixelsCorn
Worth         76.08000000         NA
Hamilton      96.32000000         NA
Kossuth       50.48087468  0.1904752
Hancock       28.54476316  0.2846382
Winnebago      0.08145147  0.3872573
Webster        5.48252687  0.4258783
Pocahontas    -8.78651636  0.5006440
Franklin     115.56683286  0.1268856
Hardin        16.39881070  0.3446977
Cerro Gordo  165.76000000         NA
Humboldt    -272.70292308  1.0603077
Wright       -59.96862032  0.5802991

Degrees of freedom: 37 total; 13 residual
Residual standard error: 18.11868
>
```

- Also the plots below shows our model is good fit.



Model for Hectareas of Soy:

```
> lm.soy <- lmList(soy_group)
> lm.soy
Call:
  Model: HASoybeans ~ PixelsSoybeans | CountyName
   Data: soy_group

Coefficients:
             (Intercept) PixelsSoybeans
Cerro Gordo    8.0900000             NA
Humboldt     -60.6714634     0.69939024
Franklin     -19.8303128     0.42699004
Winnebago     81.1814777     0.04629555
Worth        103.6000000             NA
Hamilton     106.0300000             NA
Wright         0.9008238     0.52662596
Kossuth       13.5769597     0.53856942
Hancock       51.0786087     0.28694638
Hardin         1.3200740     0.41953325
Webster      -25.6522301     0.54365580
Pocahontas   -62.4479223     0.69939996

Degrees of freedom: 37 total; 13 residual
Residual standard error: 14.16772
>
```



- The limitation with these linear models is that the dependent variable could be correlated with one or more covariates in the linear model which can be handled by multiple regression.
- In addition, these models do not consider the random factor of counties since there were a "sample" of segments for each counties.

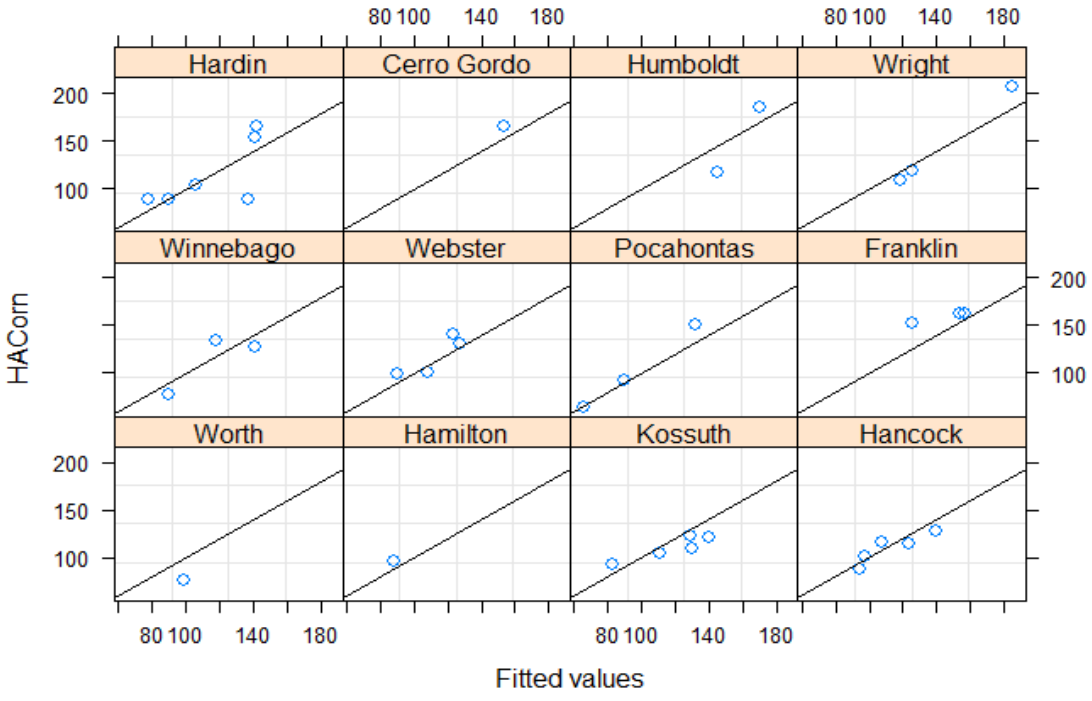
(b) The next task in this exercise was to fit a linear mixed model $y_{ij} = x^t\beta + v_i + e_{ij}$ for both crops such that segments share the same countywide random effect.

- We use the *lme* function to fit two models: *lmm.corn* for the linear mixed model of of corn and *lmm.soy* for the linear mixed model of Hectareas of soy.
- The random effects are provided by the random object.
- In the fixed model, we observe the respective standard deviations of corn and soybeans. Hectares of corn model:

```
> summary(lmm.corn)
Linear mixed-effects model fit by REML
 Data: corn_group
       AIC      BIC    logLik
  326.6529 332.8743 -159.3264

Random effects:
 Formula: ~1 | CountyName
        (Intercept) Residual
StdDev:    7.926246 17.03993

Fixed effects: HACorn ~ PixelsCorn
                Value Std.Error DF  t-value p-value
(Intercept) 5.466189 13.543455 24 0.403604  0.6901
PixelsCorn  0.387836  0.043575 24 8.900464  0.0000
 Correlation:
            (Intr)
PixelsCorn -0.961

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-2.8145038 -0.5576048  0.1739121  0.6701859  1.5489571

Number of Observations: 37
Number of Groups: 12
>
```

We can see that all of the covariates are significant with p-value < 0.05.

- For each county we compare the observed and the fitted value of the Hectares of corn.



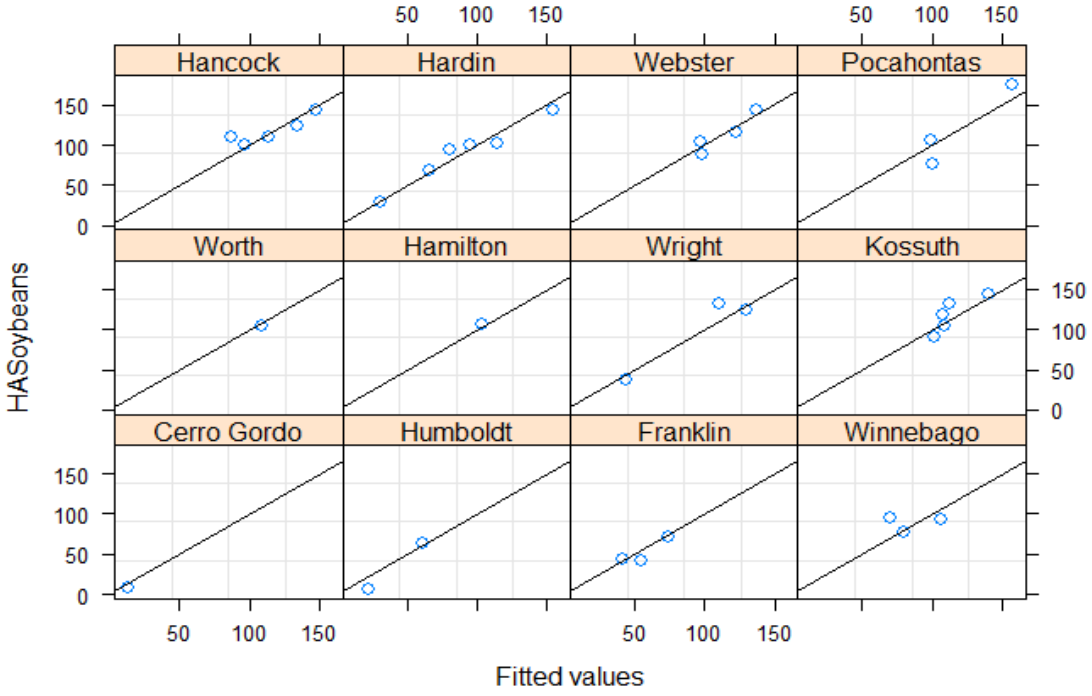
We can see that the the points are close to the straight line so we have an acceptable model for the prediction of hectares of corn.

Hectares of soy model:

```
> summary(lmm.soy)
Linear mixed-effects model fit by REML
 Data: soy_group
       AIC      BIC    logLik
  321.0191 327.2405 -156.5095

Random effects:
 Formula: ~1 | CountyName
        (Intercept) Residual
StdDev:    15.46753 13.41709

Fixed effects: HASoybeans ~ PixelsSoybeans
                   Value Std.Error DF   t-value p-value
(Intercept)    -3.822356  9.325208 24 -0.409895  0.6855
PixelsSoybeans  0.475678  0.039701 24 11.981527  0.0000
 Correlation:
                (Intr)
PixelsSoybeans -0.835

Standardized Within-Group Residuals:
       Min         Q1        Med         Q3        Max
-1.8087062 -0.5328769 -0.1997715  0.4419333  1.8973958

Number of Observations: 37
Number of Groups: 12
>
```

We can see that only the variable *PixelsSoybeans*.

- For each county we compare the observed and the fitted value of the Hectares of soy.



We can see that the the points are close to the straight line so we have an acceptable model for the prediction of hectares of soy.

(c) In order to obtain predictions for $\mu_i = \bar{x}_{ip}^t \beta + v_i$ four predictors are compared and evaluated with respect to their reliability.

- Regression predictor
- Adjusted survey predictor
- (Empirical) BLUP
- Survey predictor

An estimate for the mean squared error $MSE_{\mu_i}(\mu_i^t) = E(\mu_i - \mu_i^t)$. We create a list with the predictions using each of the above predictors and print the respective MSE for each county and both crops.

- Model: the model "Soy" or "Corn"

- We present the 4 types of estimators in the next tables:

  Regression predictor:

Regression predictor				
County	Predictor Corn	MSE corn	Predictor Soy	MSE soy
Cerro gordo	119.99	72.06	86.41	230.25
Hamilton	121.97	71.97	89.72	234.47
Worth	117.78	71.74	93.82	233.62
Humboldt	118.23	69.42	100.93	227.81
Franklin	128.88	65.35	85.63	209.21
Pocahontas	105.21	64.34	113.73	198.74
Winnebago	118.63	65.82	84.35	207.97
Wright	122.31	65.62	101.47	214.25
Webster	107.15	64.09	113.71	189.56
Hancock	127.36	63.19	90.68	198.14
Kossuth	121.29	62.27	93.50	198.02
Hardin	131.90	63.19	80.39	195.19

Adjusted survey predictor:

Adjusted survey predictor				
County	Predictor Corn	MSE corn	Predictor Soy	MSE soy
Cerro gordo	135.23	283.62	72.16	301.71
Hamilton	131.77	280.63	95.87	290.79
Worth	90.27	287.68	82.33	291.52
Humboldt	108.71	138.58	74.73	162.66
Franklin	150.43	95.66	61.38	98.69
Pocahontas	116.03	92.96	113.05	98.51
Winnebago	113.07	96.77	100.80	100.28
Wright	124.62	95.44	115.57	100.89
Webster	117.18	72.70	109.22	74.08
Hancock	121.13	59.00	101.90	64.38
Kossuth	104.38	58.09	123.08	58.61
Hardin	130.51	50.62	73.70	52.44

(Empirical) BLUP:

Empirical BLUP				
County	Predictor Corn	MSE corn	Predictor Soy	MSE soy
Cerro gordo	122.70	61.67	76.76	175.45
Hamilton	123.71	61.72	93.89	160.09
Worth	112.89	61.13	86.04	161.28
Humboldt	115.35	53.48	70.80	215.44
Franklin	137.36	43.28	49.28	282.89
Pocahontas	109.47	43.50	112.71	283.99
Winnebago	116.44	43.28	109.00	284.83
Wright	123.22	43.90	122.60	284.74
Webster	117.80	38.47	105.77	380.60
Hancock	124.12	34.44	112.90	478.34
Kossuth	112.50	33.53	152.03	470.74
Hardin	131.12	32.26	65.99	562.89

Survey predictor:

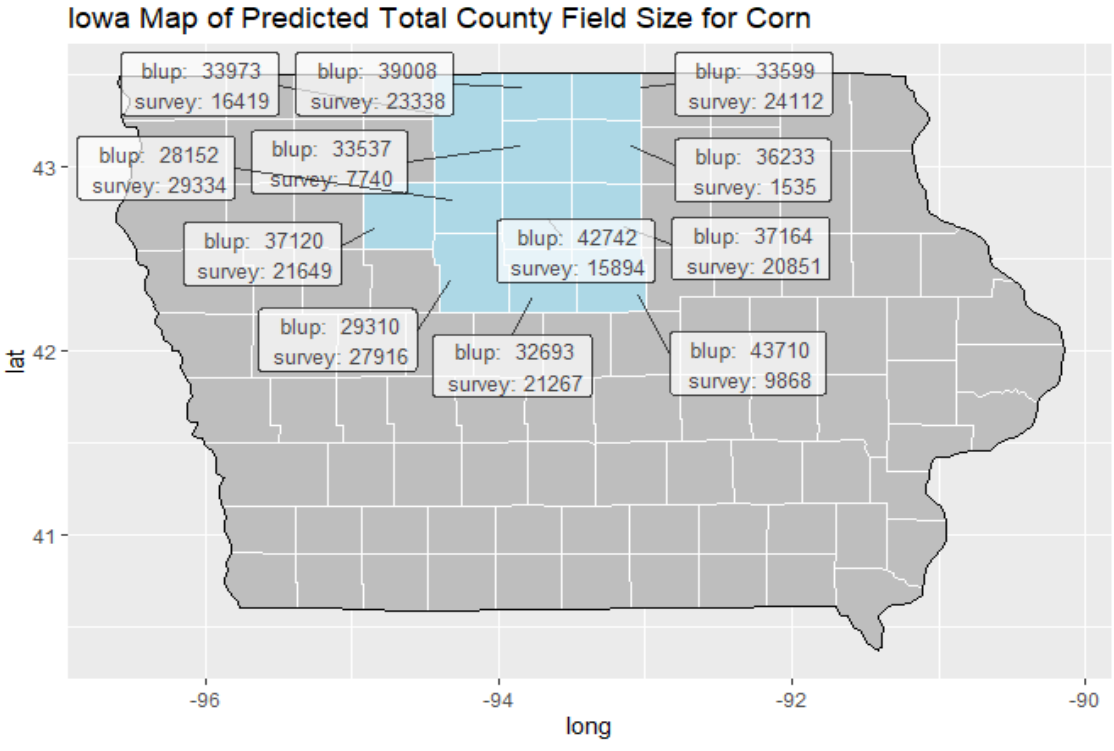
Survey predictor				
County	Predictor Corn	MSE corn	Predictor Soy	MSE soy
Cerro gordo	165.76	302.12	8.09	327.64
Hamilton	96.32	306.22	106.03	291.29
Worth	76.08	292.90	103.6	294.47
Humboldt	150.89	167.64	35.14	159.41
Franklin	158.62	97.63	52.47	97.50
Pocahontas	102.52	99.09	118.69	97.07
Winnebago	112.77	96.79	88.57	98.14
Wright	144.29	101.67	97.80	99.65
Webster	117.59	72.59	112.98	72.72
Hancock	109.38	59.81	117.47	60.27
Kossuth	110.25	58.51	117.84	58.32
Hardin	114.81	50.62	89.77	50.74

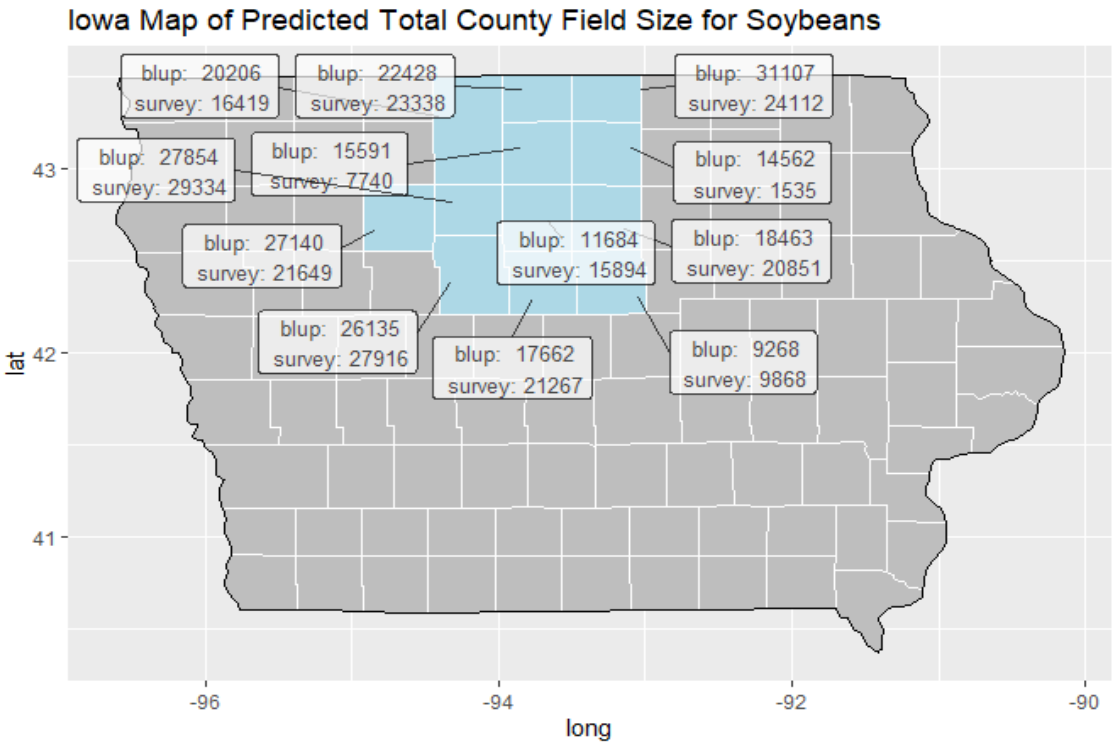
(d)

- Finally,the total county field size was estimated as the total segments multiplied by the estimators from (c).We create a data frame df_total to save these results.These results are presented in the next table:

Total predicted county field				
County	Corn Blup	Soy Blup	Corn survey	Soy survey
Cerro gordo	36232.59	14561.56	4894.27	1534.67
Hamilton	37163.87	1846.22	28934.53	20850.80
Worth	32692.98	17661.63	22032.77	21267.01
Humboldt	33537.21	15590.69	43869.76	7739.63
Franklin	43710.09	9267.84	50475.53	9868.13
Pocahontas	28151.91	27854.09	26365.93	29333.13
Winnebago	33973.49	20206.23	32903.88	16418.84
Wright	37119.98	27139.80	43470.8	21649.00
Webster	29310.33	26134.81	30829.88	27916.23
Hancock	39007.80	2242.89	34376.57	2338.18
Kossuth	33598.79	31107.08	32926.76	24112.06
Hardin	42742.30	11684.45	37426.91	15894.07

- These estimates have been plotted on the map of Iowa based on the twelve counties. See the maps below




Iowa Map of Predicted Total County Field Size for Soybeans
blup: 20206
survey: 16419
blup: 22428
survey: 23338
blup: 31107
survey: 24112
blup: 27854
survey: 29334
blup: 15591
survey: 7740
blup: 14562
survey: 1535
blup: 27140
survey: 21649
blup: 11684
survey: 15894
blup: 18463
survey: 20851
blup: 26135
survey: 27916
blup: 17662
survey: 21267
blup: 9268
survey: 9868
lat
43
42
41
-96
-94
-92
-90
long

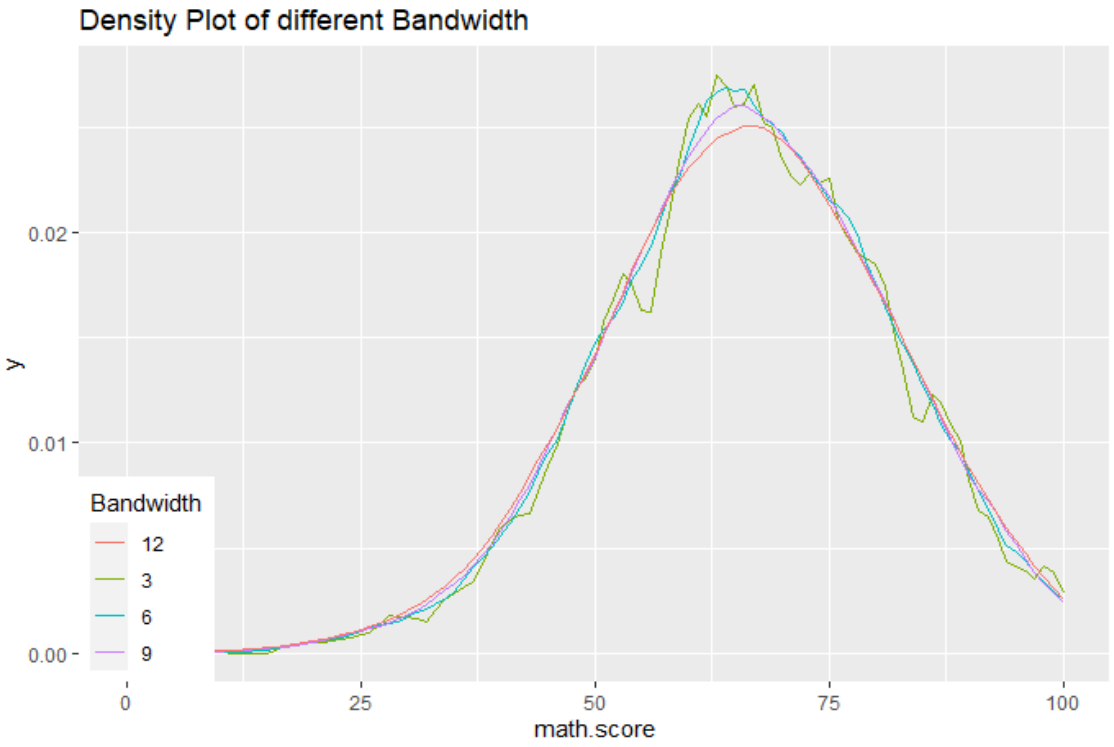
4 Kernel density estimation

(a) The first task in this exercise was to implement the Kernel Density Estimator as a function in R that depends on the sample, bandwidth and a kernel. This implemented function also incorporated the indicator function, as appropriate, of the kernel.

- We estimate 4 different bandwidths - 3; 6; 9 and 12 and save it in variables epa1, epa2 ,epa3 and epa4 respectively using R functions.

- We use the *kd* function created before using the 4 different bandwidth and fixing the kernel to Epanechnikov.

- Finally, we plot these 4 models:

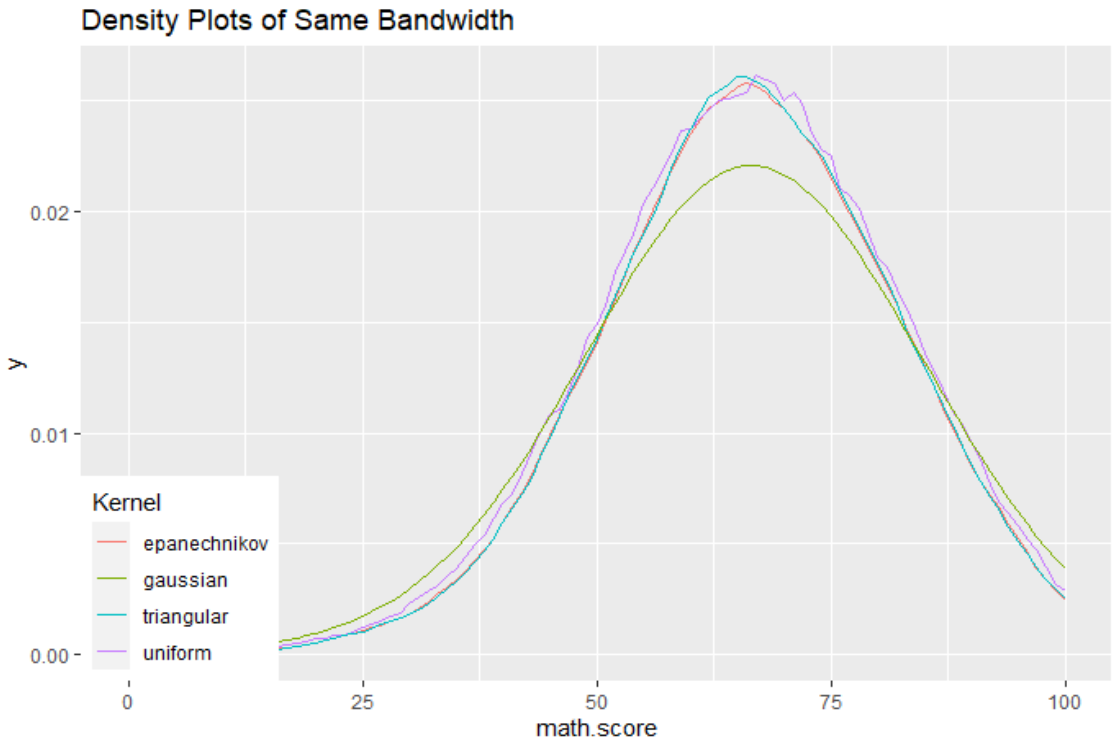


We can see that the 4 models fit properly. However it seems that the model with bandwidth of 3 is slightly erratic than the others and it might have a little bit more of bias.

- Next we fix the bandwidth to 10 since during this exercise we can compare it with our own cross validation function.

- We use the *kd* function created before using the 4 different kernels and fixing the bandwidth.Creating the vectors epa,uni,tri and gauss.

- Finally, we plot these 4 models.



We can see that the 4 models follow the behavior of the data. The most erratic behavior are the ones with Triangular and uniform kernels and the continuous ones are the Epanechnikov and the Gaussian kernel.

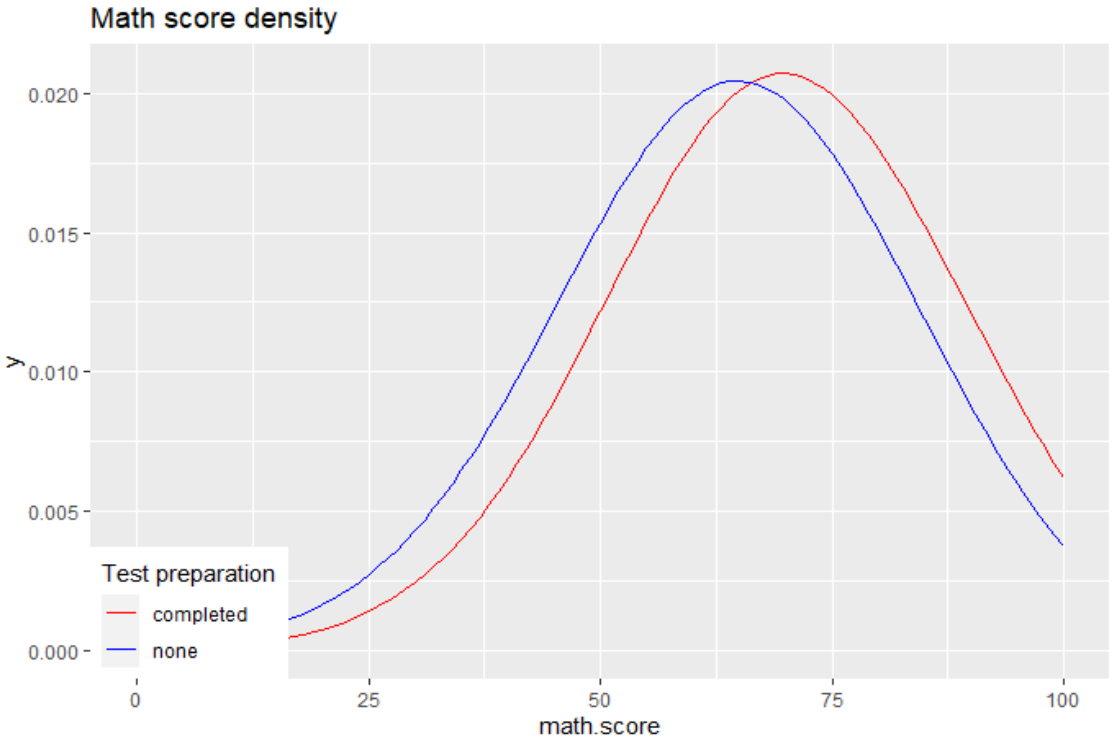
(b) The next task in this exercise was to implement cross-validation criterion to find optimal bandwidth and compare the resulting bandwidths with the ones obtained by built-in R functions bw.ucv and bw.bcv used in density for all three scores math.score, reading.score and writing.score.

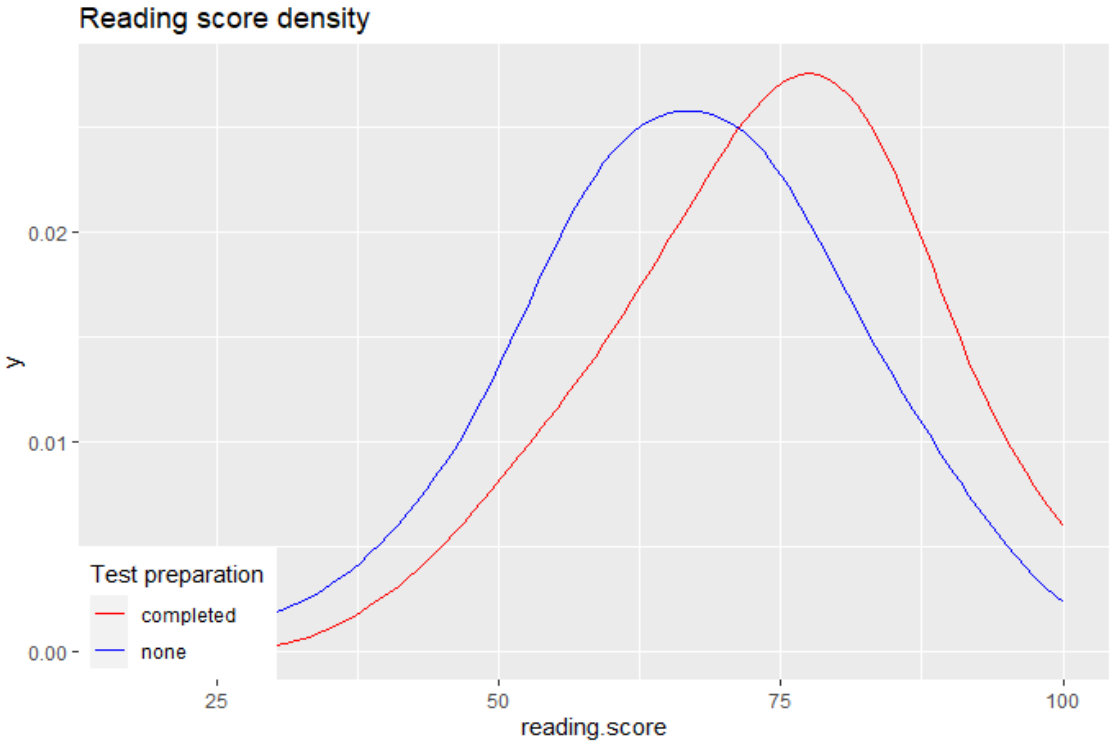
We apply our *kd* function and fixing the kernel to Gaussian since the R functions bw.ucv and bw.bcv calculate the bandwidth only using Gauss kernel. We obtain the following results in the table below:

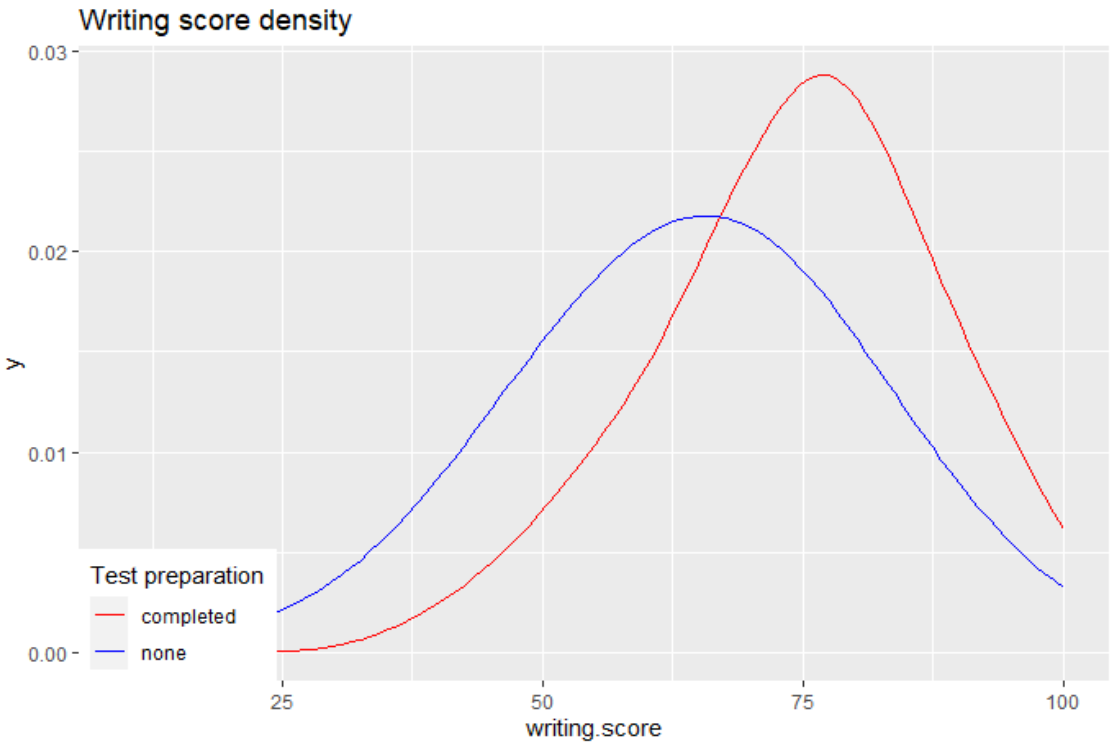
Method	math.score	reading.score	writing.score
R-function	5.51	4.46	5.49
bw.ucv	4.64	3.76	4.26
bw.bcv	4.26	4.39	4.18

From the table above we can see that our bandwidth estimations are pretty smaller than the ones calculated with *R* functions.

(c) The last task was to compare the densities of the three scores of students that did not took part in the preparatory courses and those that attended the preparatory course. The chosen bandwidth is the optimal bandwidth from the implemented R function. The figures below highlight these densities for the two categories.


Math score density
0.020
0.015
0.010
0.005
0.000
y
Test preparation
completed
none
0
25
50
75
100
math.score


Reading score density
0.02
0.01
0.00
y
Test preparation
completed
none
25
50
75
100
reading.score



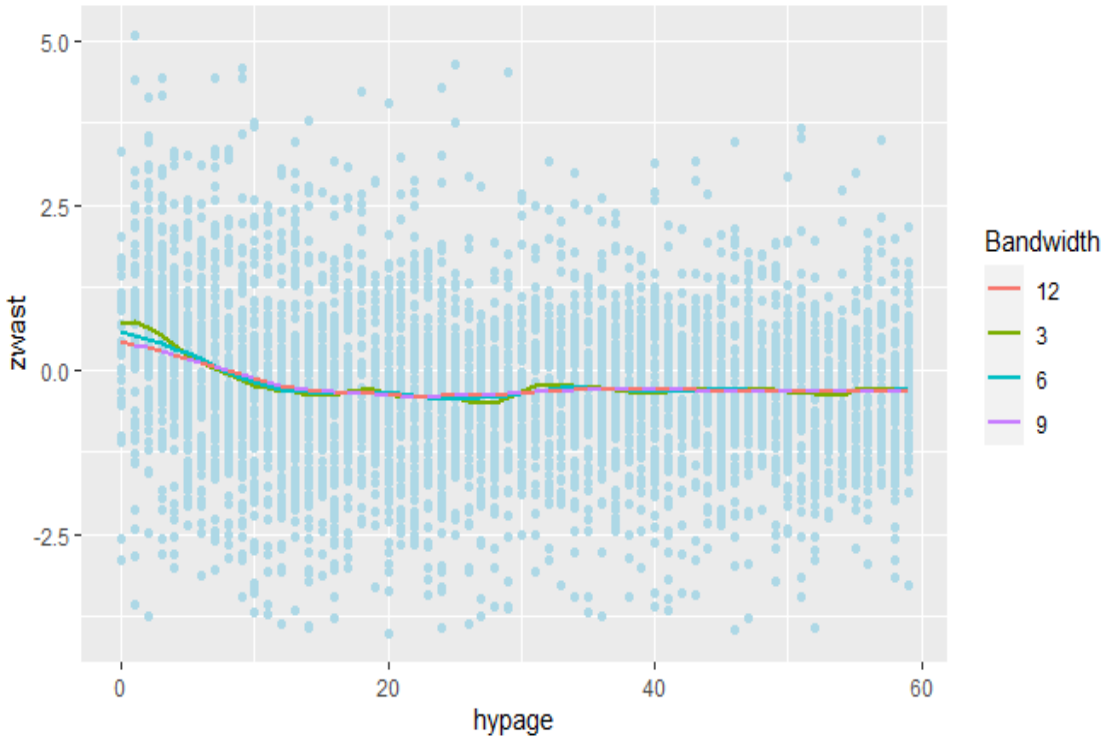
From the above plots the students performance in mathematics who completed preparation and none preparation have fairly the same density. We can also see that the density of students who completed preparation is above the none preparation in both reading and writing. It means the probability of having better score in reading and writing depends on taking the complete preparation.

5 Nonparametric regression: local polynomials

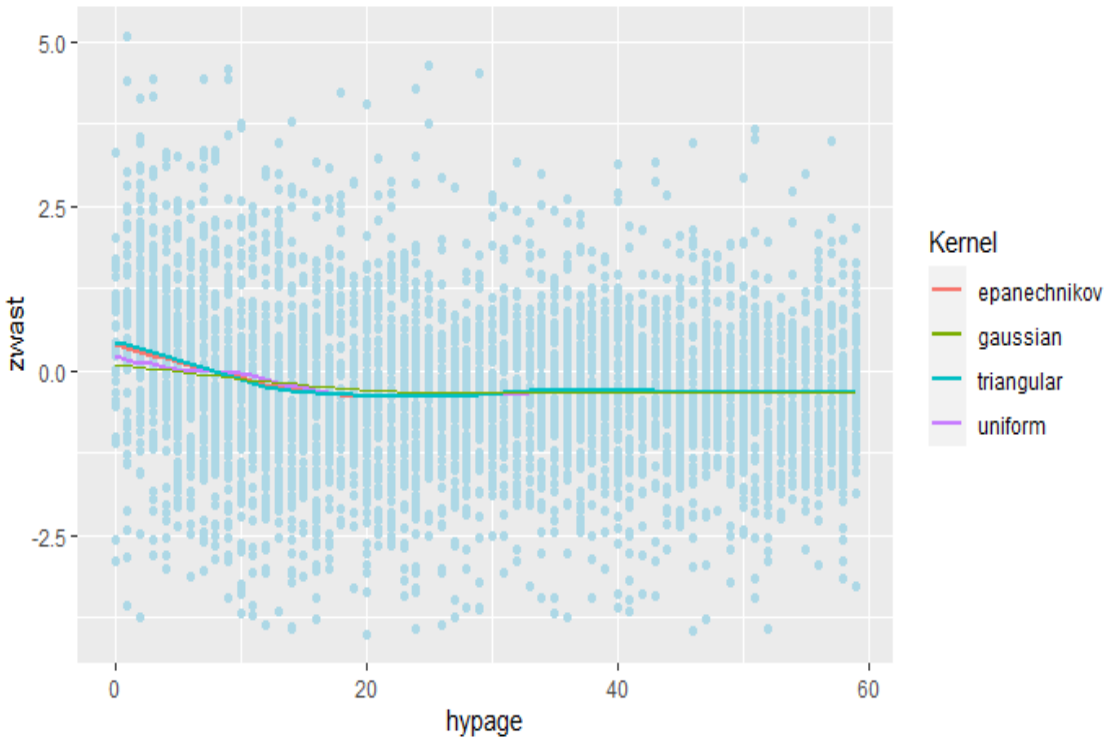
In this exercise we consider the dataset from Exercise 1 on Kenyan children data.

(a) The first task was to write an R function that calculates a local polynomial fit and depends on the response, covariate, bandwidth, polynomial degree, kernel and the number of derivatives.

- The we implemented the R function called *kernel* fixing X= hypage and Y= zwast.
- We estimated 4 different bandwidths and fixing the polynomial degree to one..
- We use the *kernel* function created before using the 4 different bandwidth and fixing the kernel to epanechnikov.
- Finaly, we plotted these 4 models:



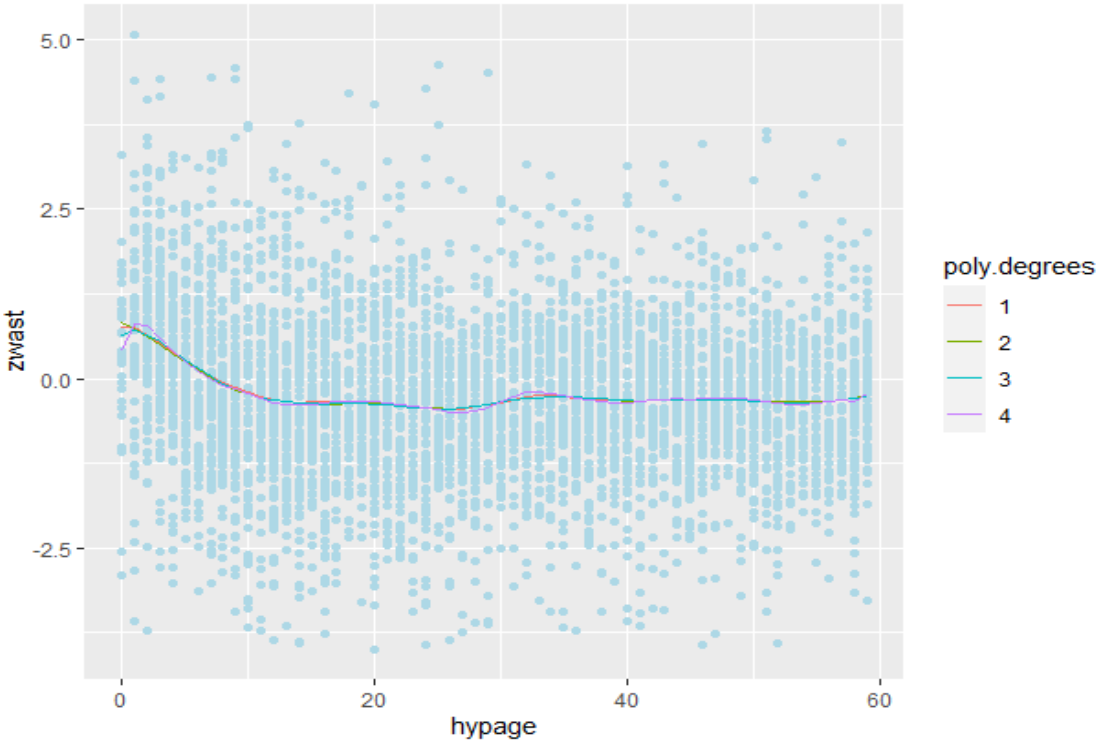
- We choose the most reasonable bandwidth and fix the polynomial degree to 1,. We fit 4 models varying the kernel densities and fixing the bandwidth(bw=10).We then plot all fits on same plot.



We can see that the 4 models fit the data properly

(b) The second task of this exercise was to write a function that calculates the optimal bandwidth with Generalized Cross Valiidation (GCV).

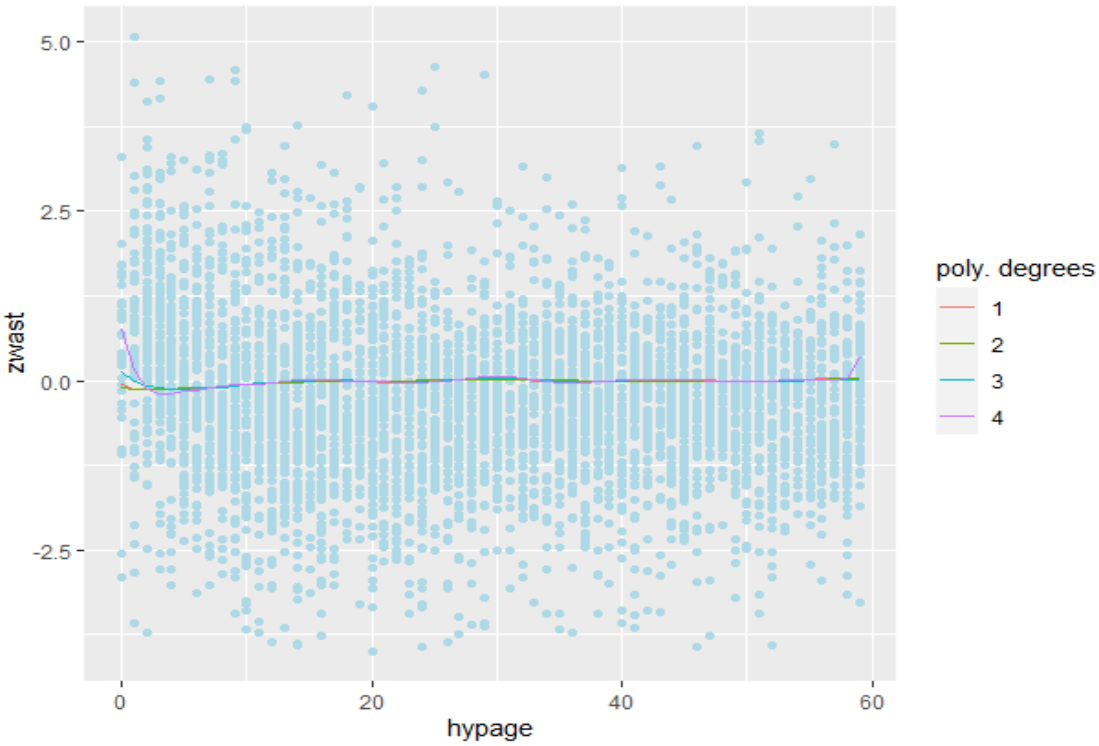
- We implement a function that calculates the GCV, then with used Epanechnikov kernel.(Refer to the code on this exercise)
- We obtained GCV-bandwidth to estimate f using polynomial degrees from 1 to 4.
- We fit four models using the implemented $local\_poly\_fit$ and save it in the variables f9, f10, f11 and f12 for each polynomial degree. This model uses the bandwidth already estimated in first part of the exercise and plot all four fits putting the curves on the same plot.



- We can see that the bandwidth estimated are reasonable since the model follow the behavior of the sample but we can see that the smoothest model is the one from polynomial degree 1 as compared to others which is less smooth. Moreover, we can see that at the boundaries there arss signs of over-fitting(for polynomial degree 4), or under-fitting (for polynomial degree 2). We can therefore argue that the polynomial with degree 3 with the corresponding optimal bandwidth would be more preferred.

(c) The last part of the exercise was to use function *localpoly.reg* of library *NonpModelCheck* to calculate the first derivative of the function of zwast with the GCV-bandwidth and polynomial degrees from 1 to 4. And then plot all four derivative fits putting the curves on the same plot.

- We created the variables derv.1,derv.2,derv.3 and derv.4 to save the 4 models using the *localpoly.reg* function.
- Refer to code on this exercise.

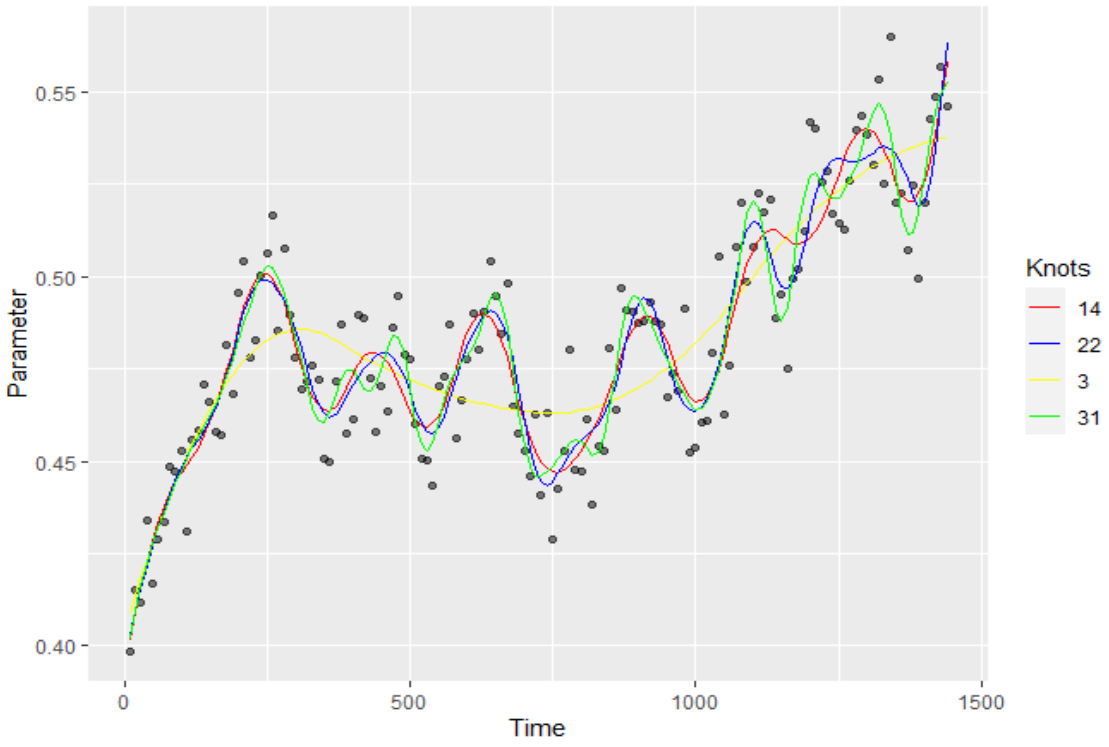


From the plot above we observe differences on the boundaries because the curve start different zwast values but the red line is almost constant since it is the model of the first derivative of a polynomial of degree one.The models for polynomial degrees larger than 2 follow a similar distribution. Since the derivative function get closer to zero after 2 years we can say that the z-score is improving very slowly.

6 Nonparametric regression: regression splines

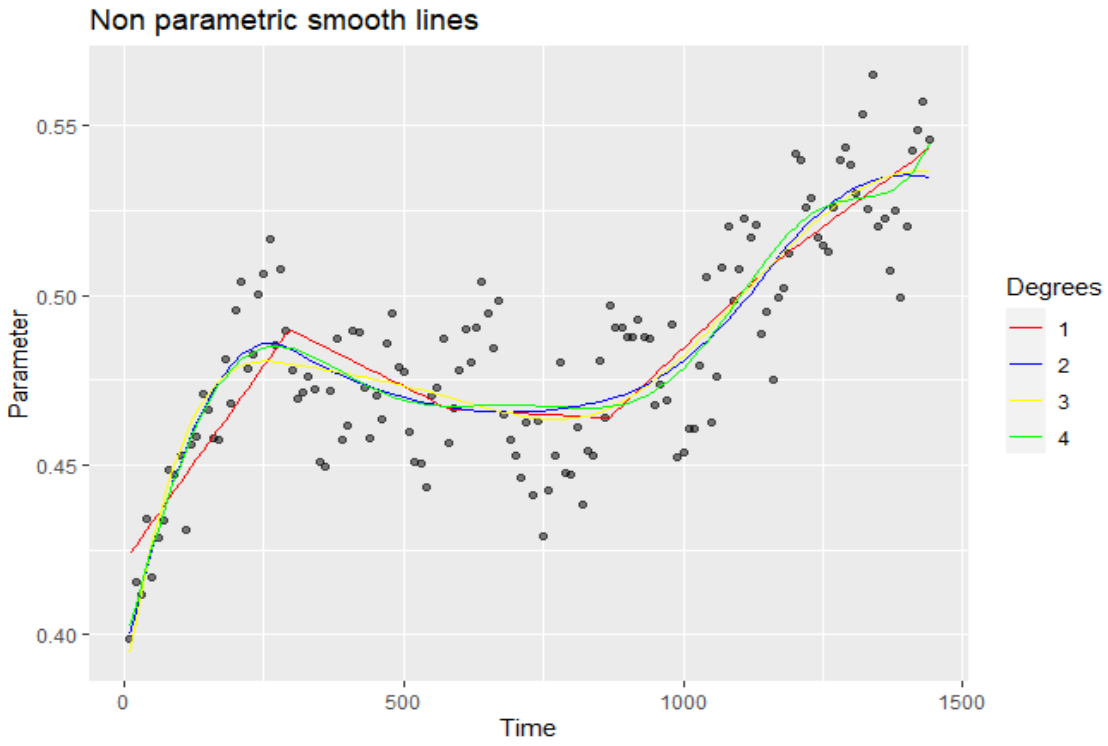
(a) In this exercise we used *stemcells.txt* dataset which contains 144 observations of the order parameter of a living stem cell observed every 10 minutes over 24 hours

- We implemented R function *sp.reg* which depends on covariate, response, predicted response and spline degree which was fixed to 2.Refer to the code related to this exercise.
- We calculated regression fit f with different *knots*.



From the plot above we observe all models underfit the data, which may be as result of small number of knots.

.
- We now fix the number of knots to 4 and using splines of degree from 1 to 4 and obtain the plot thus each model represents a polynomial degree as seen the next plot:



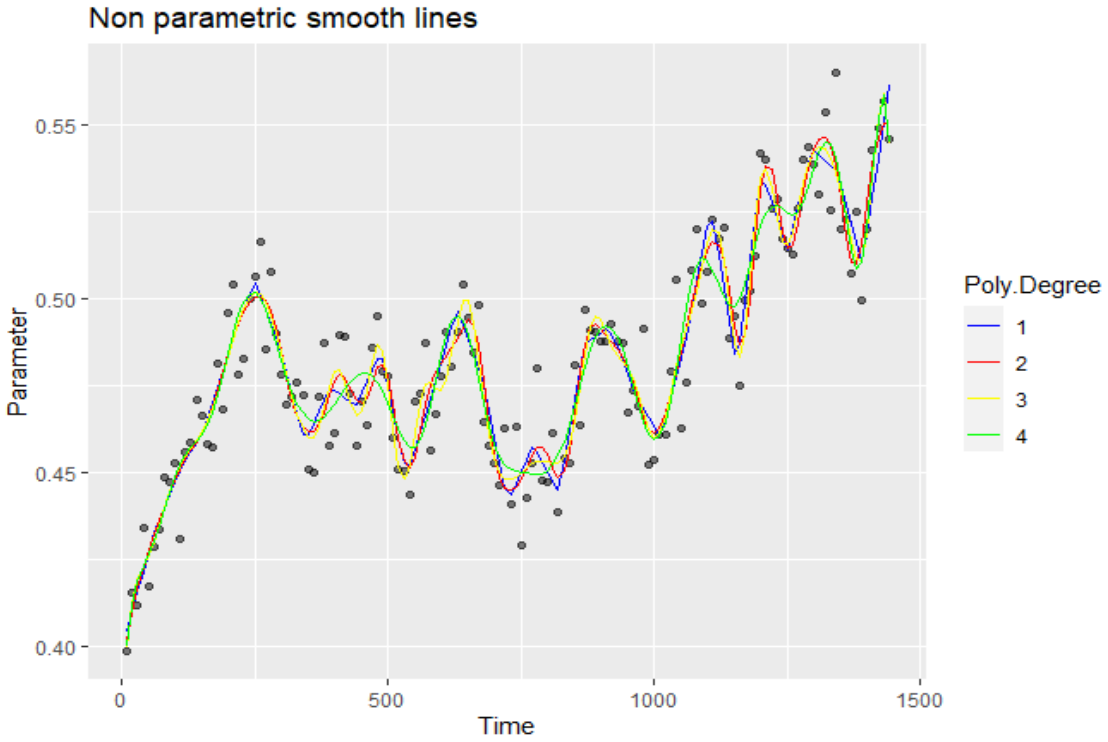
The models follow the distribution of the sample but we can see that the best fit are the models with degrees 1 and 2. Using the principle of the model with the minimum possible degree, we take the model with polynomial degree 1.

(b) The next task in this exercise was to write a function that estimates the optimal number of equidistant knots with Generalized Cross Validation (GCV).

- First we ignored that the data are dependent.
- Secondly, we created a function *gcv_knots* whose result is GCV.
- Finally, we create a function *gcvk* whose results are the the optimal number of equidistant knots and the plot of GCV.
- We apply the *gcvk* to 4 models with degrees from 1 to 4 to estimate the optimal number of knots and we save it in the variables $gcvk1,...,gcvk4$

Spline degree	1	2	3	4
GCV knots	29	33	35	21

We use the previous results to estimate 4 models using the implemented function *sp.reg* and we plot it:

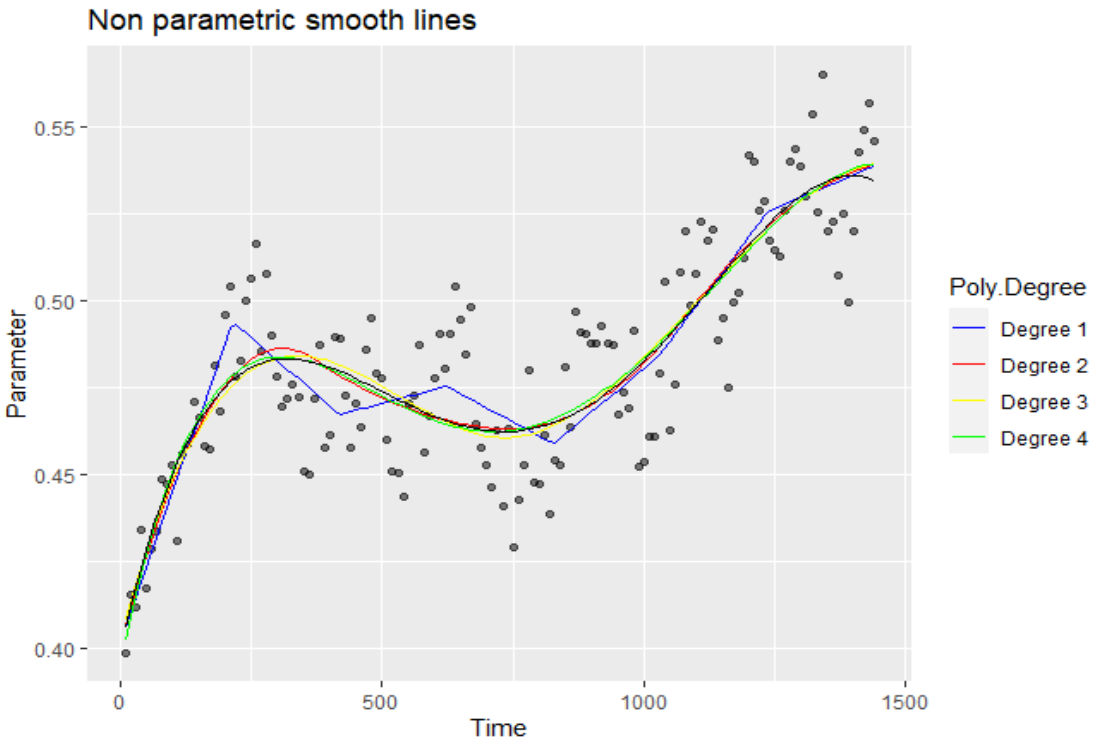


We can say that the estimators have been estimated properly since all the models follow the behaviour of the sample.

(c) Now we incorporate our knowledge about dependence in the data into regression spline estimation and GCV criterion. And we update the functions for regression splines and GCV so that they take into account that the errors ϵ_i follow and autoregressive process of order one.

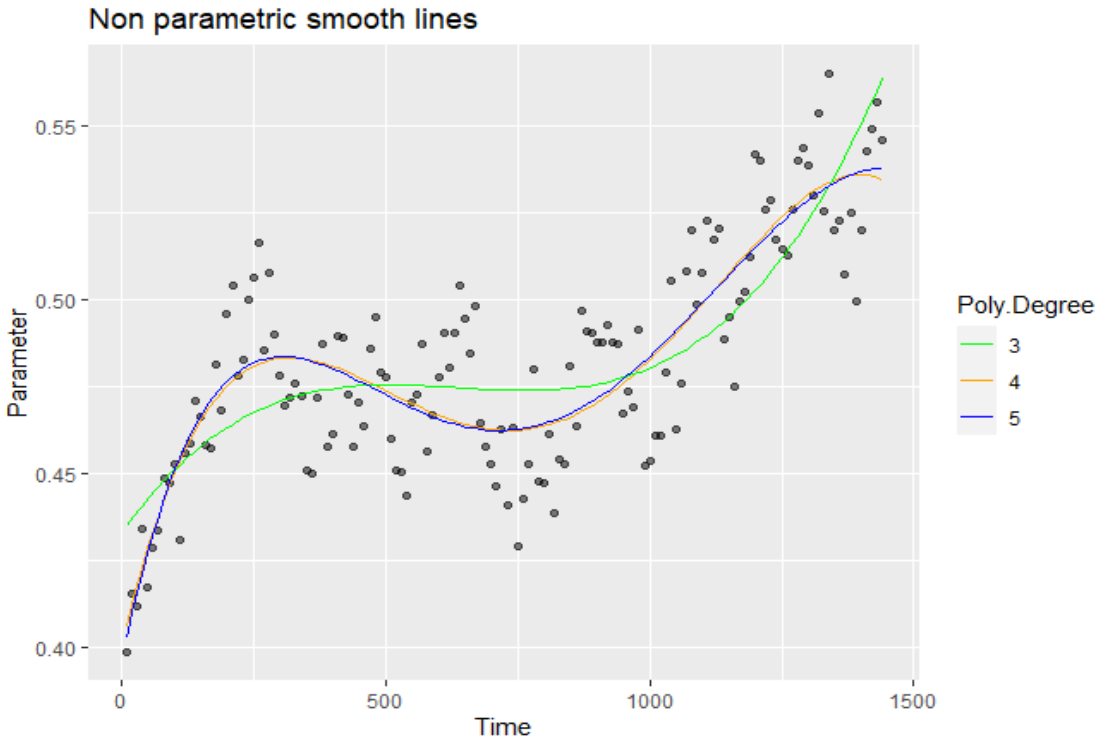
- Refer to the code on this exercise.
- The results is presented in the table below:

Spline degree	1	2	3	4
GCV knots	6	3	1	1



We can see that the 4 models fit properly the data.

. Now we the fits with the number of knots obtained with this updated GCV criterion and spline degrees from 1 to 4. and apply the same procedure in (b). We then plot resulting estimators.



We can see that the non parametric polynomial model of degree 3 is underfitted but best fits are 4 and 5 degree.

7 Analysis of big data with partial least squares

The data used for this exercise can be find on the page *myPersonality.org*.There are three files:

users.csv: contains psychodemographic user profiles
likes.csv: contains anonymised IDs and names of Facebook Likes
$users - likes.csv$: contains the associations between users and their Likes

(a) The first task in this exercise was to read all three data sets into R and update the users-likes matrix adding two columns. The first column in the ith row s contain the row number in the users matrix, where the ith user of users-likes appears. The second column in the ith row contain the row number in the likes matrix, where ith Like of users-likes matrix appears.

- Data users (data.users) contains 9 columns. The first column is the id of the user (userid) and the other 8 are psychodemographic variables. We show the first 5 rows.

```
users <- read.csv("users.csv")
head(users)
                          userid gender age political  ope   con   ext   agr   neu
54f34605aebd63f7680e37ffd299af79      0  33         0 1.26  1.65  1.17 -1.76  0.61
86399f8c44ba54224b2e60177ca89fa9      1  35         0 1.07  0.17 -0.14  1.49  0.30
84fab50f3c60d1fdc83aa91b5e584a78      1  36         0 0.89  1.28  0.86  1.07  0.99
f3b8fdaccce12ef6352bfad4d6052fe9      0  39        NA 0.33 -1.01 -0.33 -0.68  0.92
8b06ea5e9cb87c61da387995450607f7      0  31        NA 0.15  0.47  1.17 -1.01 -0.32
7a2ec9c1de4ec137367e66ad759ec848      0  38        NA 1.26  0.76 -0.46 -1.76 -0.76
```

- Data likes (data.likes) contains 2 columns. The first column is the id of the like (likeid) and the second one is the name of the event it is liked. We show the first 5 rows.

```
> likes <- read.csv("likes.csv")
> head(likes)
                            likeid                                                name
1 3c1636c878e6eb2acfd00c6b61086e38                                 REIGN by Paul Gibson
2 feca46ddb8ef04f86172ace0cb7e004c                  Cupcake Wishes & Birthday Dreams
3 b65f46d64c688fe98bdbcf93a76a71fc    Yo tambiÃ©n me rei de la caÃda de otro jejeje
4 9c5c8bb82d2cd46fbd7582f944fe370e Abraham Joshua Heschel Day School- Alumni Network
5 2d82fa84ad79b085dc516dde154327a2                            Kennesaw Farmer's Market
6 0a7a01c82143347fc4703faa4e7f415c                                   Karlsruher SC
>
```

- Data users-likes (data.users.likes) links data users and data likes, it contains two columns. The user id (userid) and the like id (likeid).

```
> users.likes <- read.csv("users-likes.csv")
> head(users.likes)
                             userid                             likeid
1 71bc7c0901488aec6d30f0add257e7c5 3c1636c878e6eb2acfd00c6b61086e38
2 978ab8e90c4d6ad1a48ef5c973b62f4d feca46ddb8ef04f86172ace0cb7e004c
3 85123b0e358907725cf19a2cb0ec3983 b65f46d64c688fe98bdbcf93a76a71fc
4 ce110562b3e2f7e5cad3775b32d9caa5 b65f46d64c688fe98bdbcf93a76a71fc
5 8188d20745471273fa69ba44a5b28473 b65f46d64c688fe98bdbcf93a76a71fc
6 e9c621f322640e5f3b3842d19b38aaa8 b65f46d64c688fe98bdbcf93a76a71fc
>
```

- To add the columns to the users-likes data, the function *cbind* was used so we can link the row number of (userid) from the data.users and the row number of (likeid) from the data.likes to the users-likes data.

```
> head(users.likes)
                             userid                             likeid users_row likes_row
1 71bc7c0901488aec6d30f0add257e7c5 3c1636c878e6eb2acfd00c6b61086e38     59353         1
2 978ab8e90c4d6ad1a48ef5c973b62f4d feca46ddb8ef04f86172ace0cb7e004c     36793         2
3 85123b0e358907725cf19a2cb0ec3983 b65f46d64c688fe98bdbcf93a76a71fc     30998         3
4 ce110562b3e2f7e5cad3775b32d9caa5 b65f46d64c688fe98bdbcf93a76a71fc     50637         3
5 8188d20745471273fa69ba44a5b28473 b65f46d64c688fe98bdbcf93a76a71fc     72224         3
6 e9c621f322640e5f3b3842d19b38aaa8 b65f46d64c688fe98bdbcf93a76a71fc     79675         3
>
```

. With the help of these two new columns and function sparseMatrix from the library Matrix build a M with 1 at position i; j, if user i made Like j.

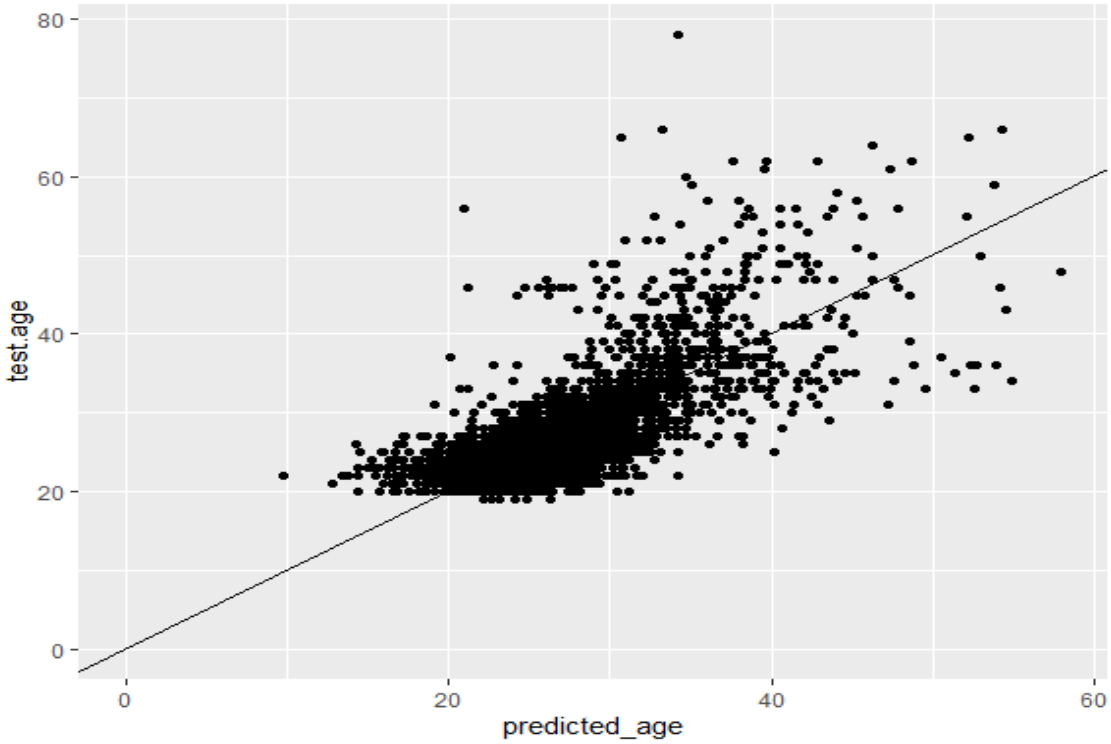
- Basically we have to create a *users.likes.M* whose rows are the users and the columns are all possible events to be liked. For example the first two rows and two columns of the matrix:

(b) The next task in this exercise was use Partial Least Squares (PLS) to find a model that allows to predict the user's age based on Likes (s)he made. First we split the *users.likes.M* matrix, as well as the corresponding age vector (found in users.csv) into a test and training set. For this, sample randomly two thirds of all rows to include into the training set and the rest will be the test set. To ensure comparability of the results set set.seed(1122) before sampling.

- We split the data in 3 equal parts using *sample* function.
- The first two thirds are stored as training set, data to be modeled.
- The last third is stored as test set, data to make comparison of the accuracy.

. On the training set fit PLS regression models with age as a response variable and with up to 50 PLS components.

. The next task was to plot the values predicted age by the PLS model of dimension d_{opt} on the test set against corresponding age values from the test set.



We can see that the points are placed around the straight line therefore implies this model is accurate.

(c) The next task in this exercise was investigate which Likes predict the age best, using the best predictive model you identified in (b).And to find 6 Likes that have the largest positive effect on the age and 6 Likes that have the largest negative effect on the age.

- Top 6:

  In the screen shot below we showed the top 6 likes related to the 20 components that predict best the age of people from the sample.

```
> x
                                                  Family Feud
                                                    0.6047770
                                                  The Hangover
                                                    0.6439420
                                                   True Blood
                                                    0.6512165
I am so old I have actually dialed a rotary phone before!
                                                    0.6934034
                                        Small Business Saturday
                                                    0.7151148
                                                          PBS
                                                    0.8391009
> |
```

- Bottom 6:

  In the screen shot below showed the bottom 6 likes related to the 20 components that predict "worst" the age of people (male or female)

from the sample. For example, we can see that for the first component, the bottom 6 likes are related to factors associated to more general likes since this can be expected for a different range of age.

```
> y
                                        Skittles
                                      -0.6716678
                                    Rise Against
                                      -0.3973009
                               Humans of New York
                                      -0.3943828
                                       Lil Wayne
                                      -0.3906041
I HATE WAKING UP FOR SCHOOL!!!!!!!!!!!!!!!!
                                      -0.3831586
                                       Duck Tape
                                      -0.3765741
>
```

(d) The last task in this exercise was to rerun the analysis removing users that made less than 60 Likes and Likes that have less than 120 users in (a). How does this influence the results

We rerun all the procedure again with the new parameters and we present also the top 6 and bottom 6.

- Top 6:

  In the screen shot below we showed the top 6 likes related to the 20 components that predict best the age of people from the sample.

```
> m
      Amazon.com   In-N-Out Burger Victoria's Secret
      0.01140979        0.01200327        0.01248825
           Weeds       Foo Fighters        Fight Club
      0.01269661        0.01303523        0.01413732
>
```

- Bottom 6:

  In the next table we show the bottom 6 likes related to the 20 components that predict "worst" the age of people.

```
> n
                                              SpongeBob SquarePants
                                                        -0.01979459
                                                              Music
                                                        -0.01824395
                                                          Lady Gaga
                                                        -0.01749860
                                                             Skittles
                                                        -0.01722567
Mom, mom, mommy, ma, mom, mom, ma, ma, mommy, mommy... WHAT!!... hi!
                                                        -0.01688124
                                  Worst. Idea. Ever. [pause] Let's do it.
                                                        -0.01628083
> |
```

In general with this change, the likes of the components remain the same.

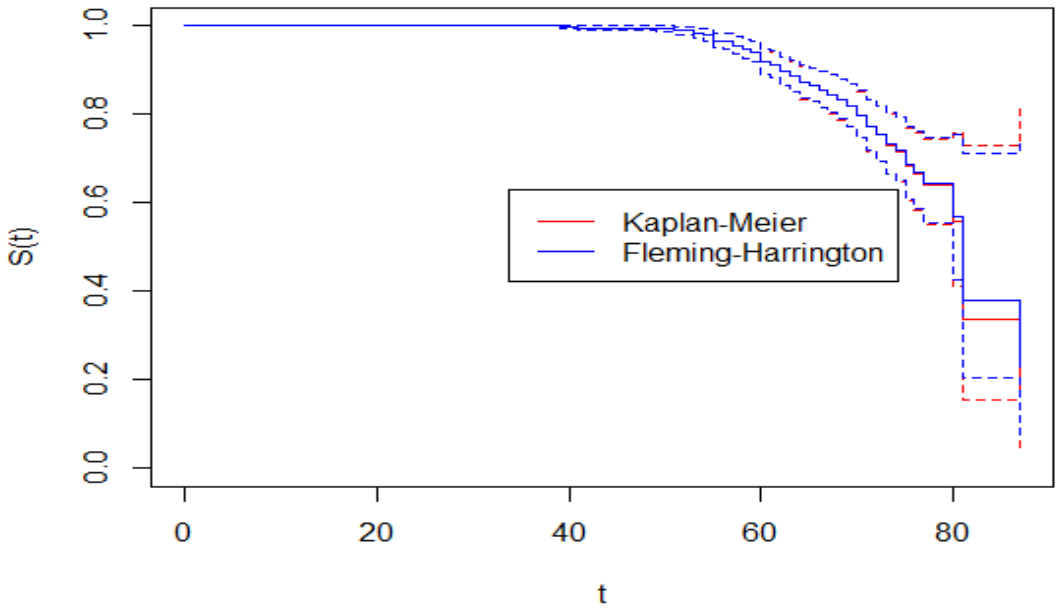
8 Survival Analysis

In this exercise dataset *Thoracic.txt* which can be found on **UCI Machine Learning Repository** was used.

(a) The first task in this exercise was to compute nonparametric estimators of the survivor function: Kaplan-Meier and Flemming-Harrington and then plot them on one plot together with 95 % confidence bands.

- To fit these non parametric models that function $surv\_fit$ was used from the survival package in R. Taking the variable Risk1Y as the censored flag. The table with the probability survival and confidence band can be shown using $summary(fit.km)$ function:

- Plot of both survival functions.

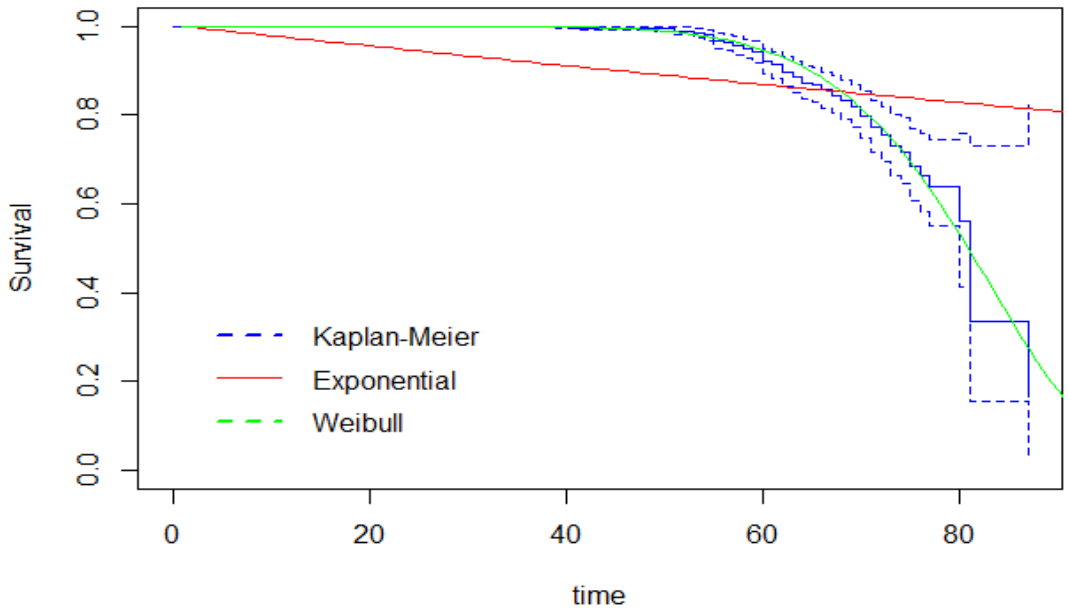


We can see in the plot above that both survival functions mostly the same however for a person above 80 years old, Fleming-Harrington estimate a survival probability greater than that of Kaplan-Meier.

.

. We fit the Exponential and Weibull models to the data and plot the Kaplan-Meier estimator together with the corresponding confidence bands and both parametric estimators for the survivor function on one plot.

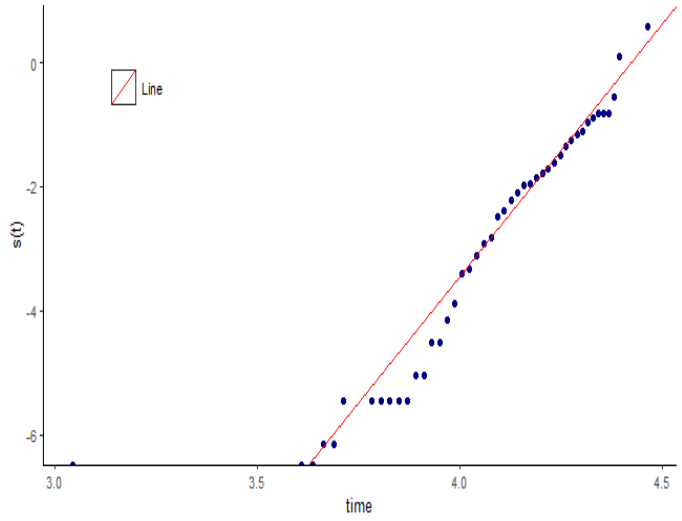
- Survival plots:



We can see in the plot above that both estimators are decreasing,but the exponential estimator for survival model is linear, while the Weibull fit has almost the shape as the nonparametric fit Kaplan-Meier. Hence we can argue that exponential model is a not a good model for the data, while the Weibull estimator would be appropriate, that is realistic.

. Weibull model:

. To check whether the Weibull model is adequate for the data we used the plot below.

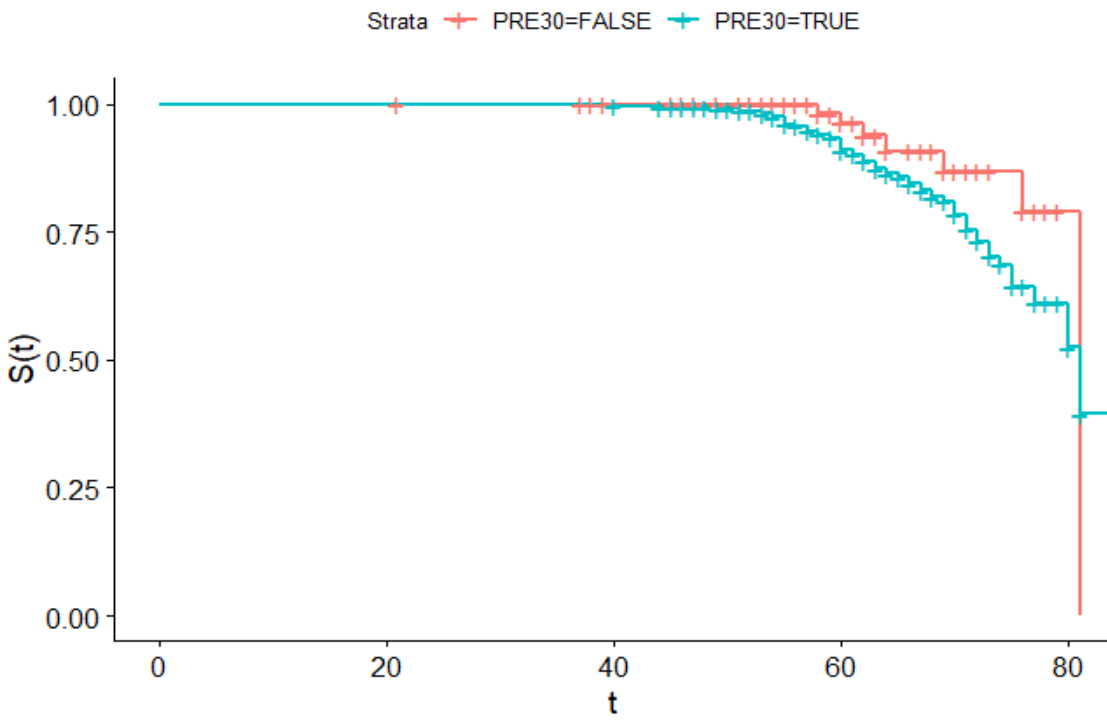


We can see that the points follow line pattern,therefore we can argue that the Weibull model is more appropriate for the data.

(b) The next task in this exercise was to split the data into smokers and non-smokers and examine the proportion of smokers in the sample. Also for each group we compute Kaplan-Meier estimators, plot together with the corresponding confidence bands.

. From the split, smokers make up 82% in the proportion and non-smokers 18%. For each group, the Kaplan-Meier estimators are computed, and plotted together with the corresponding confidence bands in the plot below.

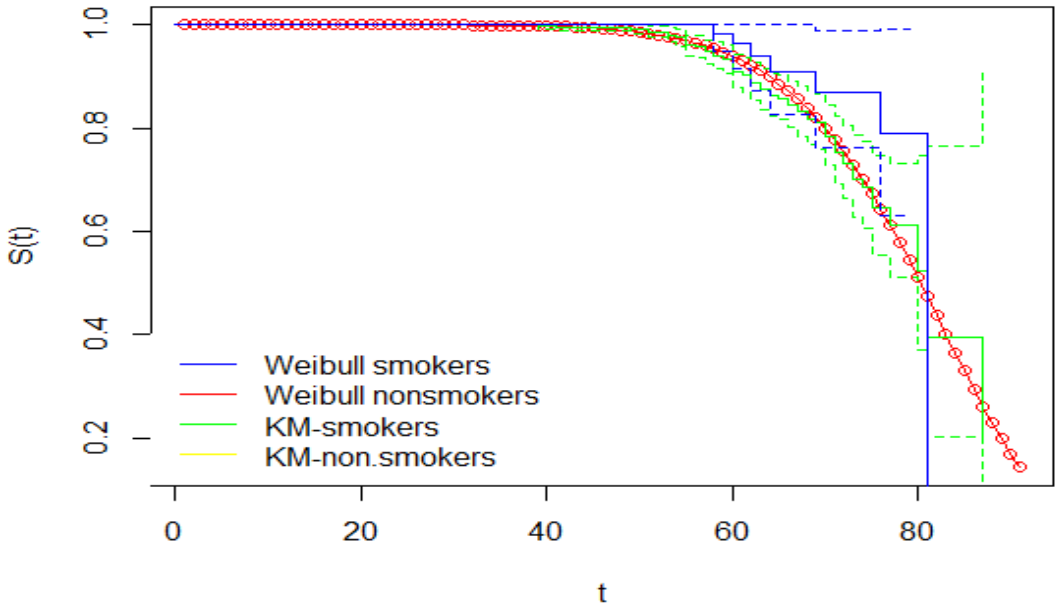
- Kaplan-Meier estimators fit:



- We can argue from the plot above that the two groups do not share the same survival function.

. Finally we plot Weibull model to both groups of smokers and nonsmokers.We then plot the resulting parametric estimators for the survivor function together with the corresponding Kaplan-Meier estimators.

- Smokers and Non-smokers:



- We can see in both smokers and non-smokers,that the Weibull model generate a good fit for the two groups models.

9 Bootstrap

(a) In this exercise we simulate a sample $(x_1, ..., x_n)$ from the Weibull distribution with the scale parameter $\lambda = 13$ and shape parameter $k = 1$. The variance of a Weibull distributed random variable is given by $\sigma^2 = \lambda^2[\Gamma(1 + 2/k)] - \{\Gamma(1 + 1/k)\}^2]$,while the median $x_{med} = \lambda\{log(2)\}^{1/k}$. We are interest in building confidence intervals for σ based on a statistic $\hat{s}^2 = (n-1)^{-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ and for x_{med} based on the sample median.

We can see that the standard deviation σ=13 and median x_{med}=9.01

- The first task was to build two-sided bootstrap percentile confidence intervals for σ and x_{med} at the significance level $\alpha = 0.95$.

  For the estimation of the bootstrap percentile the algorithm from Lecture 9 was used with the following specifications:

  - The expressions of $\hat{s}^2$ has been used as an estimator of σ^2 and the median as an estimator of x_{med} as defined above.
  - The function *rweibull* was used to generate n observations following a weibull distribution.
  - This previous steps were iterated.
  - The bootstrap confidence intervals is calculated using the quantile function to each vector *bootmedian* and *bootsd* . The lower bound of the interval using $\alpha/2$ quantile and the upper bound of the interval using $1 - \alpha/2$ quantile.

- We then use M Monte Carlo samples to estimate the coverage probability of both confidence intervals as well as also the average interval length.

  - In order to estimate the coverage probability the procedure in (a) should be repeated M times and the coverage probability $p_c(\theta)$ is estimated as: $\hat{p_c}(\theta) = \frac{\sum^M \mathbb{1}_{CI_b}(\theta)}{M}$, this might be understood as the ratio of the number of success of the parameter θ lies into the bootstrap confidence interval over the total M samples.
  - The algorithm for this consider the confidence intervals calculated in (a) and the variables *success.sd* and *success.median* count the number of times when the parameters σ^2 and x_{med} lie in their respective confidence intervals. Then:

    $\hat{p_c}(x_{med})$ = succes.median = 0.940
    $\hat{p_c}(\sigma)$ = succes.sd = 0.852

  - It means that in all the M monte carlo samples the parameter lies within its interval this occurs because we have use the exact

distribution density under the *rweibull* function and also this method is conservative since we have not applied the correction yet.

- There are now M confidence intervals for each parameter. Taking the length of each confidence interval we will have a vector with M lengths for the parameter σ called *bootsd* and a vector with M lengths for the parameter *xmed* called *bootmedian*.
- Finally, we take the average for each vector to have estimate the average interval length for each parameter:

  $avg.length$ = succes.median= 5.13
  $avg.length$ = succes.sd= 6.03

- We then change the values for n = R = 1000 and n = 100, R = 5000
  - (n=R=1000) , Under this case we have:

    $\hat{p_c}(x_{med})$ = succes.median = 0.946
    $\hat{p_c}(\sigma)$ = succes.sd = 0.933
    $avg.length$ = succes.median= 1.61
    $avg.length$ = succes.sd= 2.21
    These results show that the confidence interval decreases when the sample size n increases , this holds since for larger sample sizes in average the observations get closer to their parameter expectation.
  - (n=100, R=5000) , Under this case we have:

    $\hat{p_c}(x_{med})$ = succes.median = 0.937
    $\hat{p_c}(\sigma)$ = succes.sd = 0.850
    $avg.length$ = succes.median= 5.07
    $avg.length$ = succes.sd= 6.03

    These results show that the confidence interval is pretty similar to the first results for the original conditions of the question. The only thing that has changed is R=5000 so we can notice that the interval it does not depend on R but in n. On the other hand, the method has not changed thus the coverage probabilities neither.

. We then used R function bcanon from the package bootstrap to build bootstrap accelerated bias-corrected confidence intervals both forσ and x_{med}.

- Using M Monte Carlo samples we assess the coverage probability and the average length of the confidence intervals and then compared the results and differences to bootstrap percentile confidence intervals.

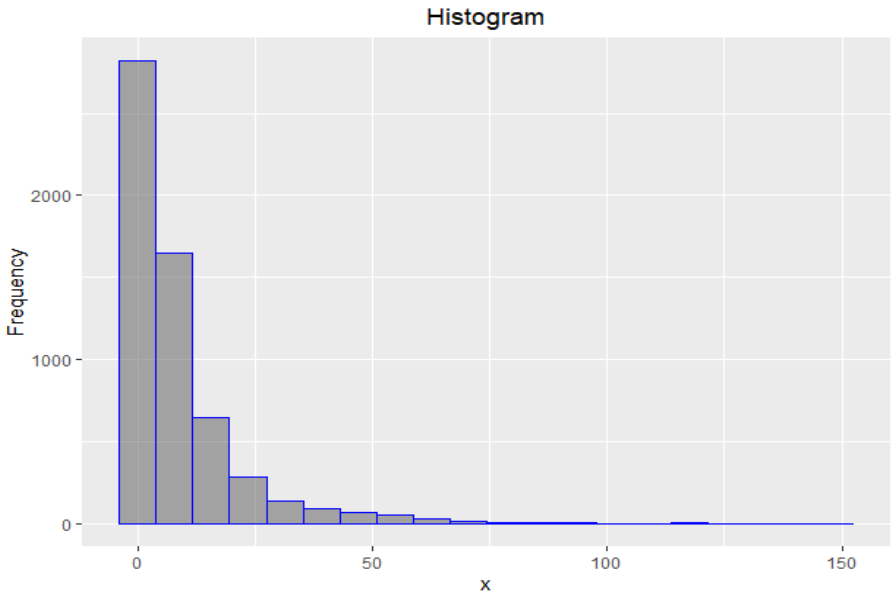
  $\hat{p_c}(x_{med})$ = succes.median = 0.951
  $\hat{p_c}(\sigma)$ = succes.sd = 0.191

$avg.length$ = succes.median= 5.05
$avg.length$ = succes.sd= 5.04

(b) The next task we used the Sleep Heart Health Study dataset $shhs2.txt$ and plot the histogram using the variable $rdi4p$ and describe the empirical distribution.

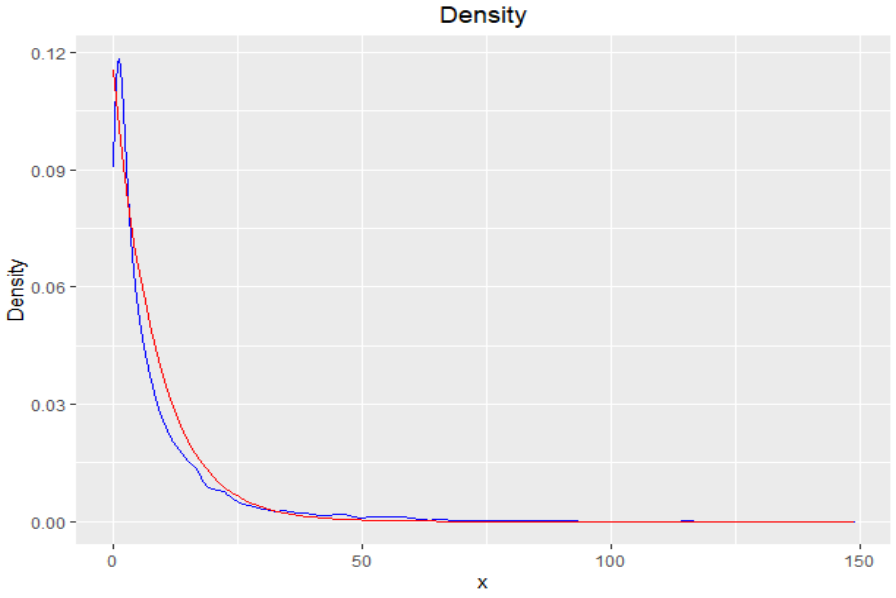
- Histogram:

  We can see in the next histogram that the variable $rdi4p$ is right long tailed and starts from zero.



- Fitting:

  In the next plot, we compare the empirical density (blue lines) and the fitted theoretical weibull (red lines) which seems to fit correctly.



(b) The last task in this exercise was to build bootstrap percentile and bootstrap accelerated bias-corrected confidence intervals for the standard deviation and median.

- Simple Bootstrap:

  For σ :

  Confidence Interval: (11.78, 13.04), length=1.26

  For x_{med}:

  Confidence Interval: (3.95, 4.42), length=0.49

- bootstrap accelerated bias-corrected:

  For σ :

  Condifence band: (11.79, 13.11), length=1.32
  For x_{med}:

  Confidence band: (3.94, 4.42),length=0.48

  For σ it can be seen that the corrected bootstrap gives a larger confidence interval for the same α level. On the other hand, for x_{med} the confidence interval corrected bootstrap is almost the same as the simple bootstrap.

10 Random number generation

(a) The first task in this exercise was to switch the default random number generator in R to Wichmann-Hill and simulate N =1000 binomial random variables B(n = 10; p = 0:4) using three approaches: inversion method, by simulating corresponding Bernoulli random variables by inversion method and using R function rbinom

- First, we use the function *RNGkind* and change this generator to "Wichman-Hill" to simulate the observations using the next three approaches:

  Inversion method:

  - We created the vectors $uniform$ and *bins* to store the mass distribution and cumulative distribution of Binomial, $Bin(n = 10, p = 0.4)$.
  - We used the inversion method for discrete distributions using the vector *bins* created before.
  - Finally, We saved the observations in the variable *inv.bins*

  Sum of Bernoulli variables:

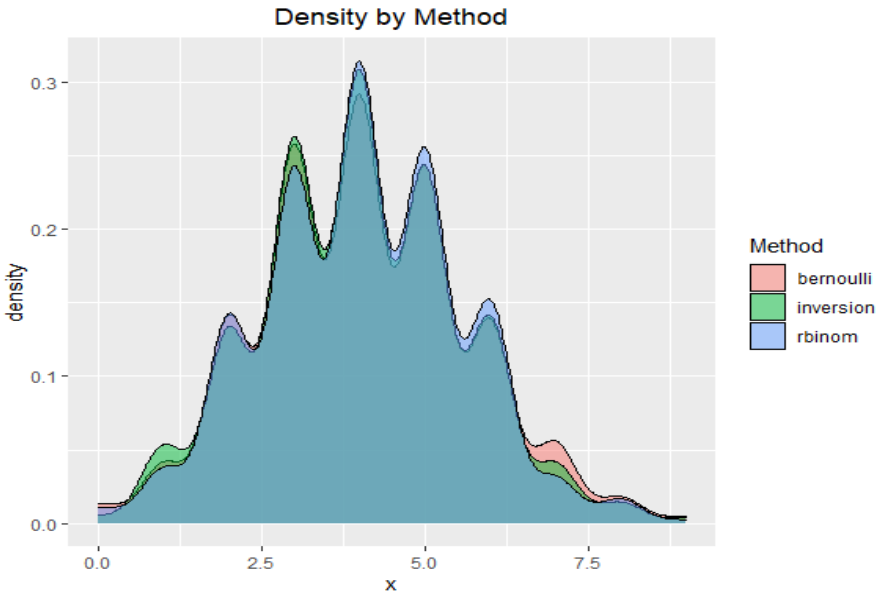
  - We used the result that the sum of n independent Bernoulli random variables is equal to a binomial random variable: Let independent random variables $X_i \sim Be(p) \Rightarrow \sum_{i=1}^{n} X_i \sim Bin(n, p)$
  - Here, we applied the inversion method to simulate 10 bernoulli trials.
  - Finally, we sum up the 10 bernoulli trials and we saved it in the vector *bin.berno*

  rbinom function:

  - We used the function *rbinom* and we saved it in the vector *bin.binom*

. The next task was to plot the empirical probability density functions of all three samples on one panel and switch the random number generator back to its default.

- We present in the next plot the simulate density distributions using the 3 different method : inversion method (inversion), sum of independent Bernoulli (bernoulli) and *rbinom* function (rbinom).



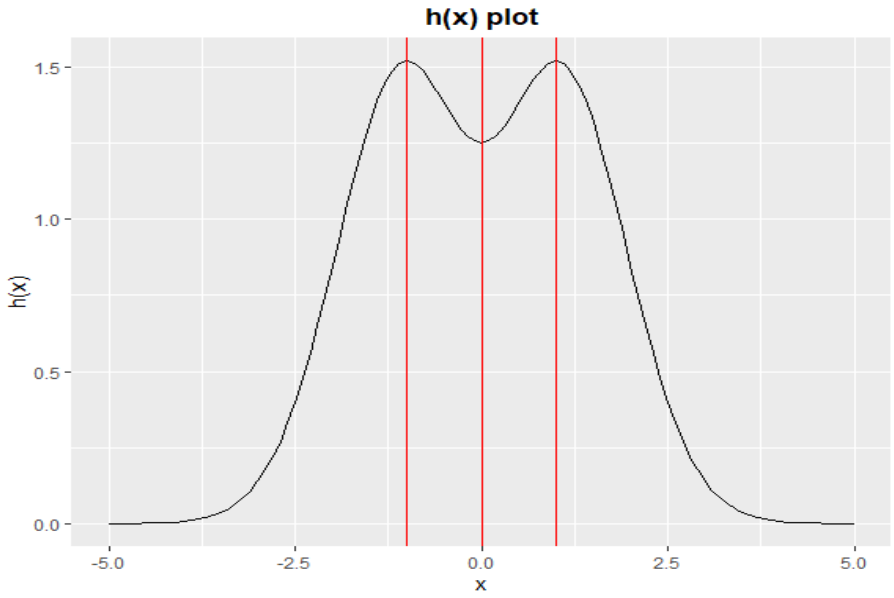
As we can see, the 3 methods give similar results but it seems that the inversion method simulate slightly more observations around p=0.4. Finally, we switch the generator method to default using $RNGkind(kind = "default")$.

(b) The aim is to simulate N = 10 000 standard normal distributed random variables with density $f(x) = 2\pi^{-1/2}exp(-x^2/2)$ using accept-reject method and a generator for uniform random variables only. As a candidate density g use the density of the standard Cauchy distribution $g(x) = \{\pi(1+x^2)\}^{-1}$

. First determine the best value of the constant c, such that $f(x) = cg(x)$.

- In such a case, we have to choose $c = sup_x \{f(x)/g(x)\}$. For this purpose, set $h(x) = f(x)/g(x)$ and calculate $h'(x) = 0$.
  $h(x) = \sqrt{\frac{\pi}{2}}exp(-x^2/2)(1+x^2)$, taking $u(x) = \sqrt{\frac{\pi}{2}}exp(-x^2/2)$ and $v(x) = (1+x^2)$
  $h'(x) = (u(x)v(x))' = \sqrt{\frac{\pi}{2}}-x^2exp(-x^2/2)(1+x^2)+2x\sqrt{\frac{\pi}{2}}exp(-x^2/2)$
  $h'(x) = \sqrt{\frac{\pi}{2}}exp(-x^2/2)x(1+x^2)$
  $h'(x) = 0$ gives the points $x = -1, 0, 1$ where
  $h(-1) = h(1) = 1.52$ and $h(0) = 1.25$ thus $c = 1.25$

  In the next plot we can see the behaviour of $h(x)$ and its 3 optimal points. Notice that the minimum of the 3 points is achieved where $x = 0$ as we proved analitically before.



(b) The next task was to obtain N standard normal random variables using the accept-reject method, generating Cauchy distributed random variables using inversion method.

- We use the following specifications from the lecture notes:

  - About functions and parameters: $f(x) = 2\pi^{-1/2}exp(-x^2/2)$, $g(x) = \{\pi(1+x^2)\}^{-1}$ and $c = 1.25$
  - We iterate the algorithm until we have 10 000 (accepted) standard normal observations.
  - The observations are stored, the indicator N counts the number of accepted observations and until we have 10 000 (accepted) observations.

. We then compared the estimated and theoretical acceptance probabilities.

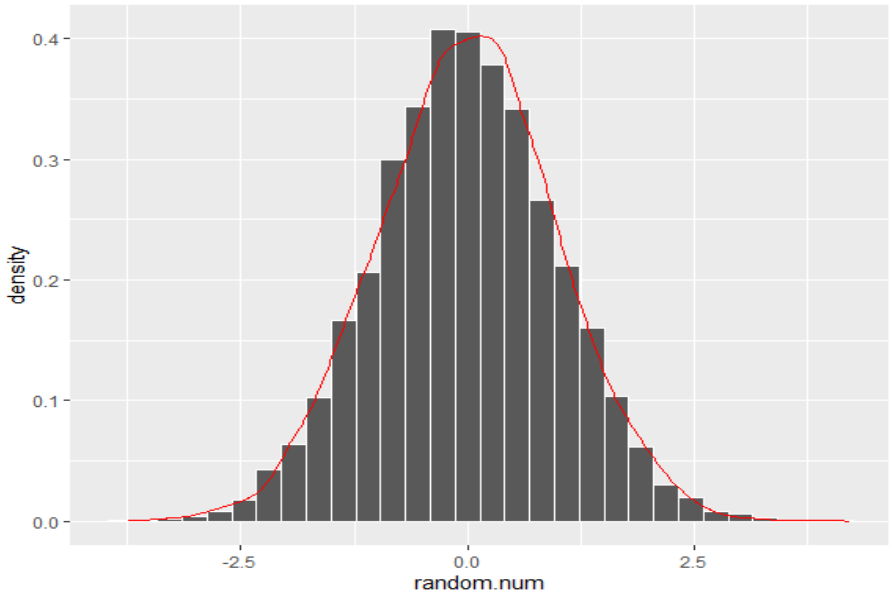
- The estimated (or empirical) acceptance probability $\hat{p}$ is calculated as $\hat{p} = N\ /totaliterations = 10000/13597 = 0.73$
- The theoretical acceptance probability p is calculated as $E[N] = 1/p = c$ then $p = 1/c = 1/1.25 = 0.8$

  We can see that these values are closed. However, this slightly less amount in the estimated probability means that it is needed more iterations than the theoretical expected one to achieve the 10 000 observations.

. We then plot a histogram of the obtained sample and add the standard normal density to the plot. Make a QQ-plot.

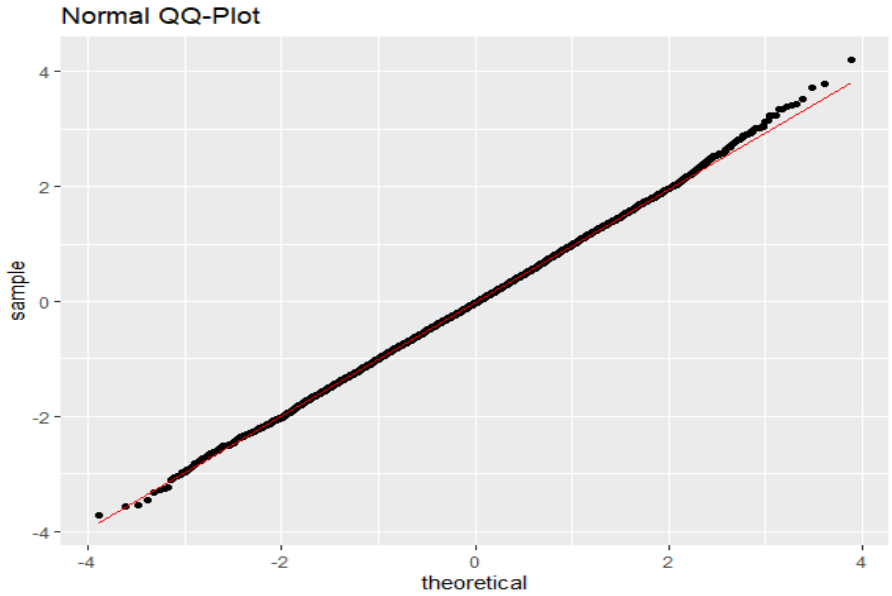
- Standard normal densities comparison:

  We can see in the plot below that the acceptance rejection method succeed in simulating normal observations since the experimental density distribution follows almost the same behaviour like the theoretical distribution.



- QQ plot:

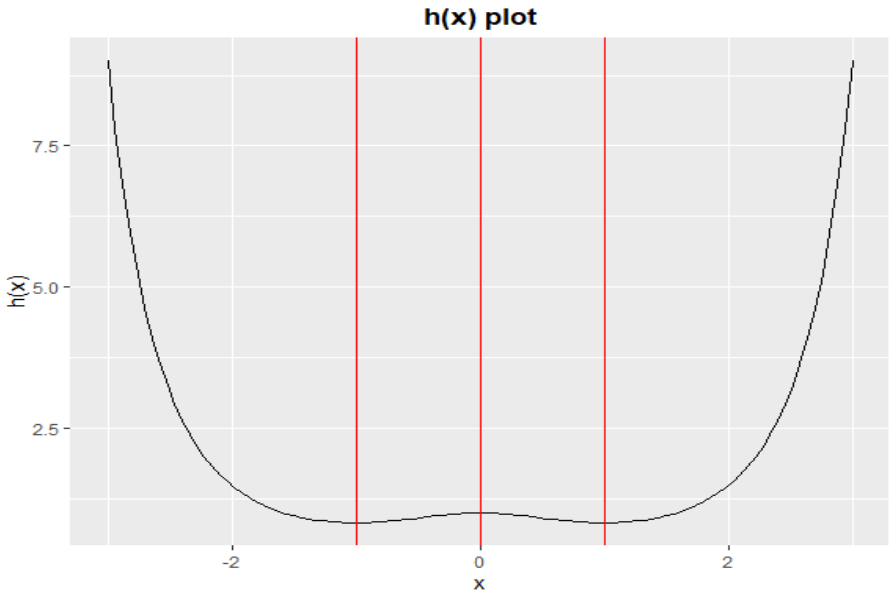
  We can see that the experimental and theoretical quantiles have pretty similar values as the plots follows the straight line.Hence the accept-reject method used to simulate a standard normal distributed sample is accurate.



. The last task in this exercise was to show that it is not possible to simulate

from the standard Cauchy density using the accept-reject method with a standard normal candidate density.

- In this case, $f(x)$ and $g(x)$ change positions and now $f(x) = \{\pi(1+x^2)\}^{-1}$ $g(x) = (2\pi)^{-1/2}exp(-x^2/2)$ thus $h(x) = \frac{(2\pi)^{1/2}exp(x^2/2)}{\pi(1+x^2)}$ whose graph is:



Even though, $h'(x) = \frac{\sqrt{\frac{2}{\pi}}exp(x^2/2)x(x^2-1)}{(1+x^2)^2} = 0$ gives us the same optimal points $x = -1, 0, 1$, these are local minimum. Furthermore, as we can see in the graph $h(x)$ is unbounded.
Analytically, $\lim_{x\to+/-\infty} h(x) = \lim_{x\to+/-\infty} \frac{f(x)}{g(x)} \overset{l'hospital}{=} \lim_{x\to+/-\infty} \frac{f'(x)}{g'(x)}$ $= \lim_{x\to+/-\infty} \frac{exp(x^2/2)}{\sqrt{2\pi}} = \infty$

We have proved that $h(x)$ is unbounded thus is not possible to choose a c and apply the acceptance rejection technique.