HW2 (PART A)

Yimo Zhang

Fall 2019

Part A

Use the data in the baseseg.csv file on kidney disease to construct a good fitting model for GFR as a function of the following potential predictors measured at baseline

1) Serum Creatinine (bascre)
2) Systolic blood pressure (sbase)
3) Diastolic blood pressure (dbase)
4) Urine protein (baseu)
5) Age
6) Gender (Sex = 1 if male; = 0 if female)
7) African-American (black)

Consider potential transformations of the outcome and variables as well as interaction terms.

Examine the fit of your model by regression diagnostics by checking model assumptions such as constancy of variance, linearity, normality and characteristics such as outliers, influence, etc).

Describe your findings in clearly written text, tables and figures. Discuss which factors are predictive and how. Show some predictive plots showing outcomes as a function of the predictors (see Papers 18 and 19 for examples).

Basic Checking

Before building a linear model, we check the structure of the whole dataset (missing values, unusual values) in order to do some elementary data cleaning.

Table 1: Summary of variables

	bascre	sbase	dbase	baseu	AGE	SEX	black	gfr
Min	0.60	91.00	50.00	0.10	15.00	0.00	0.00	0.70
1st Quantile	1.47	132.00	84.00	0.10	43.00	0.00	0.00	17.93
Median	1.98	150.00	90.00	0.98	54.00	1.00	0.00	35.56
Mean	2.27	149.42	91.48	1.80	51.95	0.65	0.06	42.59
3rd Quantile	2.80	163.00	100.00	2.63	62.00	1.00	0.00	66.00
Max	9.75	250.00	150.00	20.30	77.00	1.00	1.00	155.50
Percent of Missing values	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.33

As we can see from table above: 1) none of the variables have extreme/unusual values; 2) the only variable contaning missing values is the outcome variable GFR and about $\frac{1}{3}$ of data under that variable is missing.

Thus, we don't need to make any changes to the predictors. As for the outcome variable, even though there is considerable proportaion of data missing, however, since 1) there are no informations about the missing pattern (missing completely at random or missing at random) and neither can we check the missing pattern, and 2) the number of fully observed individuals is relatively large $1860 - 611 = 1249$ compared to the number of predictors, we choose not to impute missing values and base our analysis on the left 1249 observations.

Variable Transformation

Since the data set is clean, now we consider potential transformations of the outcome and continuous predictors. In this step, we apply transformation to variable(s) by looking at the plot between predictor and outcome. Note that we don't exclude/select predictors from the predictors pool in this step as there maybe potential confounding among predictors.

First of all, let's check the distribution of the outcome variable GFR:

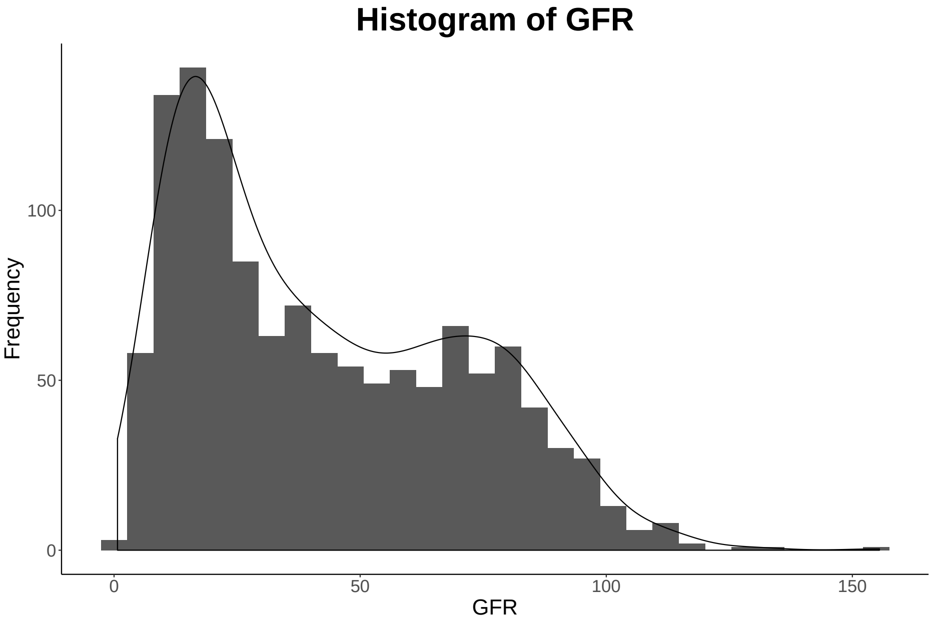


Figure 1: Histogram of GFR

As we can see, the distribution of GFR are skewed to the left, to construct normality (recall that for basic linear models

$$Y \sim N(X\beta, \sigma^2 I)$$

), we use box-cox transformation to 'normalize' GFR. Since all the values of GFR are positive, we use the transformation $\tilde{Y} = \frac{Y^\lambda - 1}{\lambda}$. The optimal λ is found by maximizing the log-likelihood of transformed data on a bunch of λ values (where we use the function *boxcox* to calculate), and it turns out that the optimal λ is 0.303.

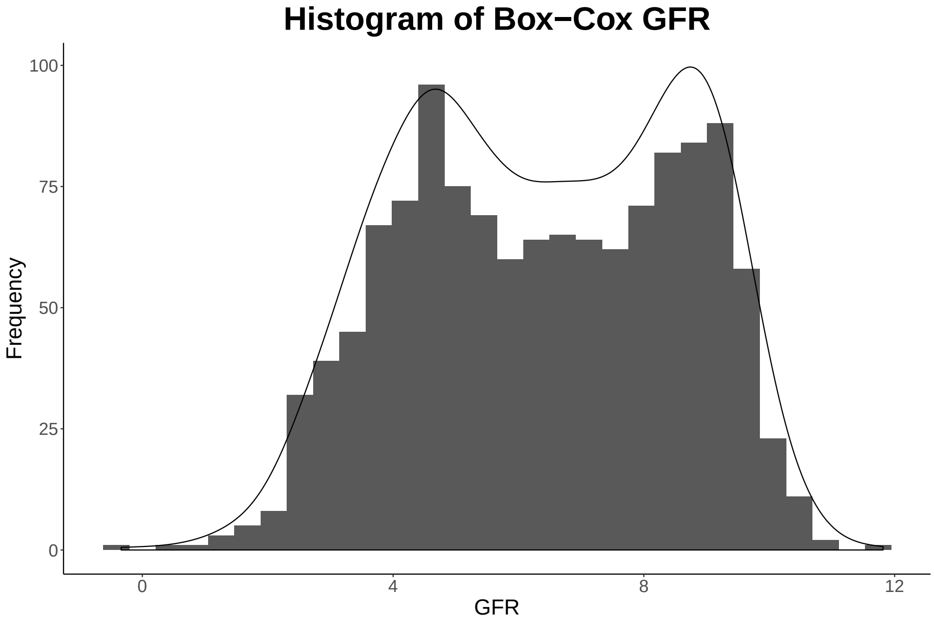


Figure 2: Histogram of Box-Cox transformed GFR

Although the figure above doesn't show a strictly normal shape (has two peaks), it does have a 'low-high-low' trend as a normal distribution, and this is a big progress from the original highly skewed distribution.

Next we look at the potential transformation of other independent variables:

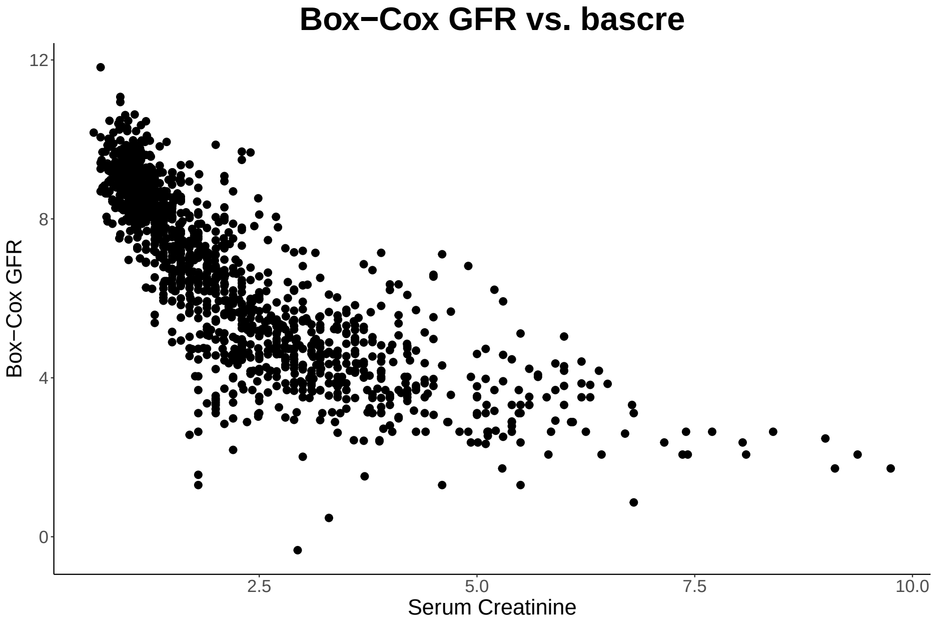


Figure 3: Point graph of Box-Cox GFR *vs.* bascre

In the figure above, we notice that most of the independent variable concentrate at the small values; also the graph has a $y = -\log x$ shape, thus we consider apply log transformation to *bascre*.

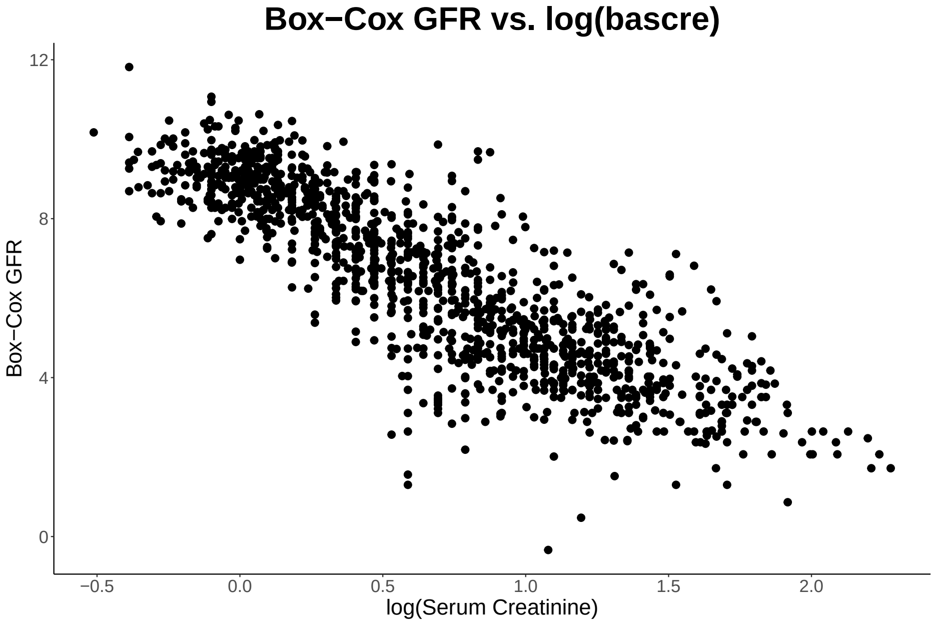


Figure 4: Point graph of Box-Cox GFR *vs.* logrithm of bascre

After transformation, we notice that there is a obvious linear relationship between the transformed bascre and transformed GFR. Therefore, we consider this transformation eligible.

Now for Systolic blood pressure:

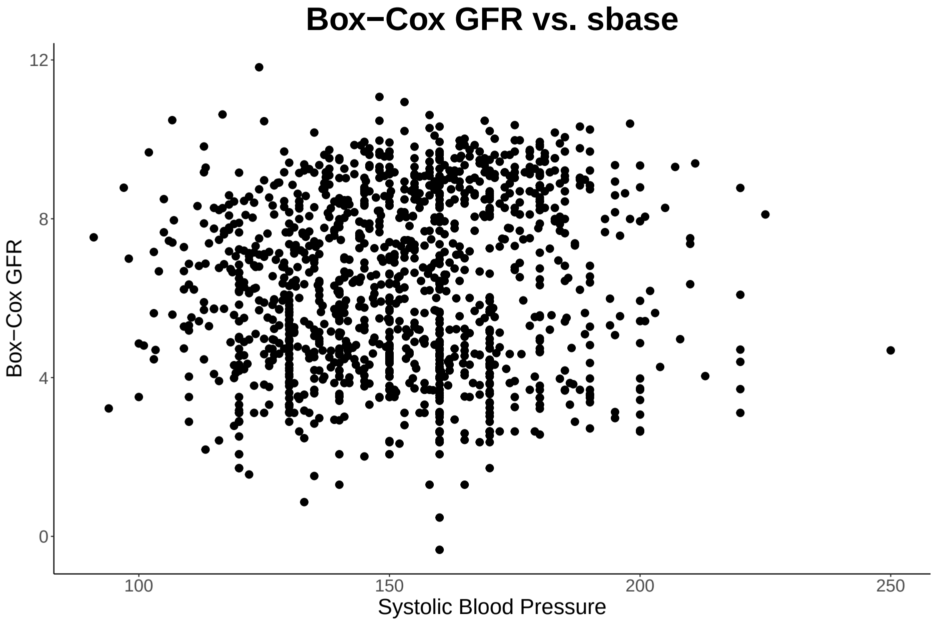


Figure 5: Point graph of Box-Cox GFR *vs.* sbase

Unfortunately, we don't see any pattern or a 'function' shape from the graph above. Although there seem to be some "vertical lines" made of some points sharing the same sbase values, however, it's not reasonable to make this variable categorical as there are considerable amount of points lying between each "vertical lines". Thus, we choose not to transform sbase.

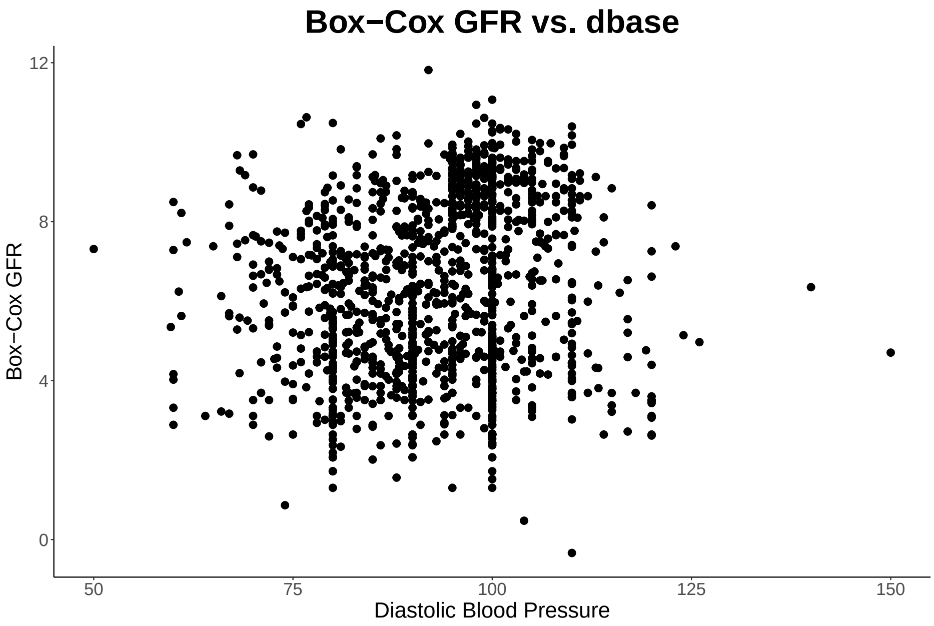


Figure 6: Point graph of Box-Cox GFR *vs.* dbase

The graph of dbase *vs.* gfr behaves pretty much the same as that of sbase *vs.* grf, therefore we don't transform dbase either. However, the similarity between these two graphs may underline some relationship between sbase and dbase (it's reasonable to think about because people who have high/low sbase tend to have high/low dbase). Moreover, if there is significant correlation between dbase and sbase, there maybe collinearity problem we have to address, but we are not worrying about this now.

For baseu:

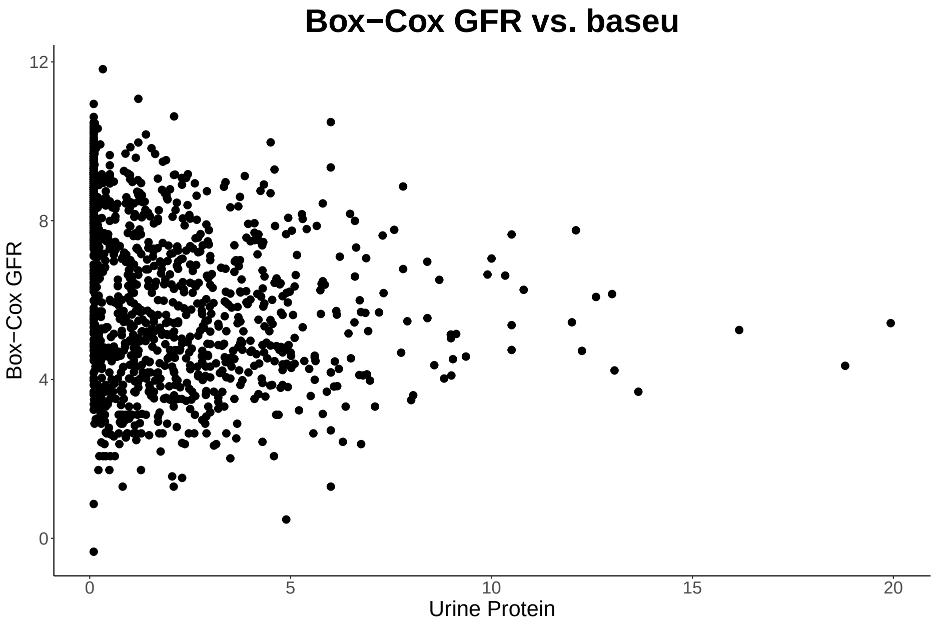


Figure 7: Point graph of Box-Cox GFR *vs.* baseu

There seems to be evidence of heteroscedasticity, but we will consider that later. Similar to bascre, baseu concentrate at smalle values, thus we apply logrithm transformation to it first.

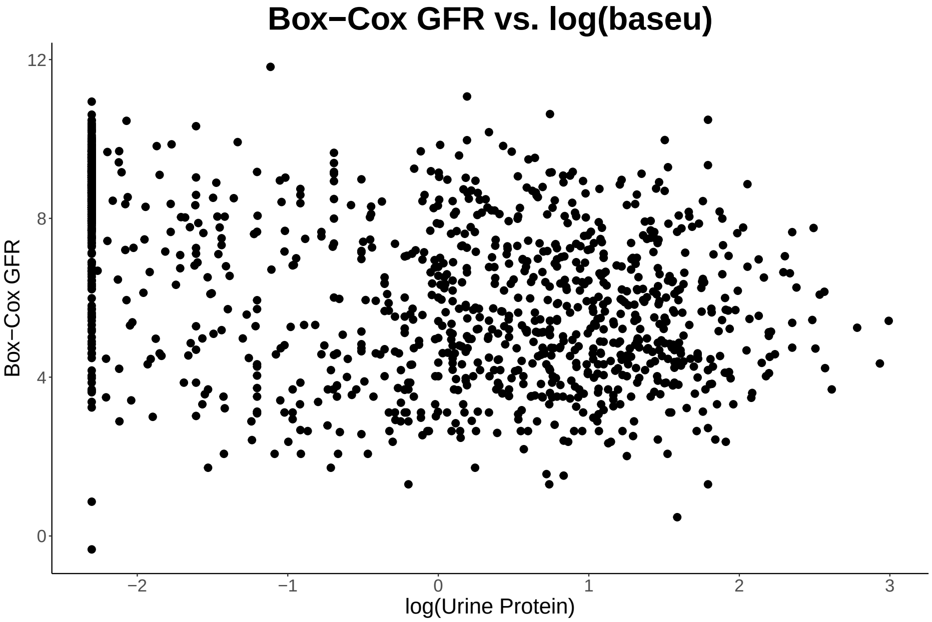


Figure 8: Point graph of Box-Cox GFR *vs.* logrithm of baseu

After the transformation, the heteroscedasticity problem seems to disappear (thus we will keep this transformation). However, there is no obvious relationship between $\log baseu$ and gfr as the points are pretty much uniformly distributed.

Finally, for AGE:

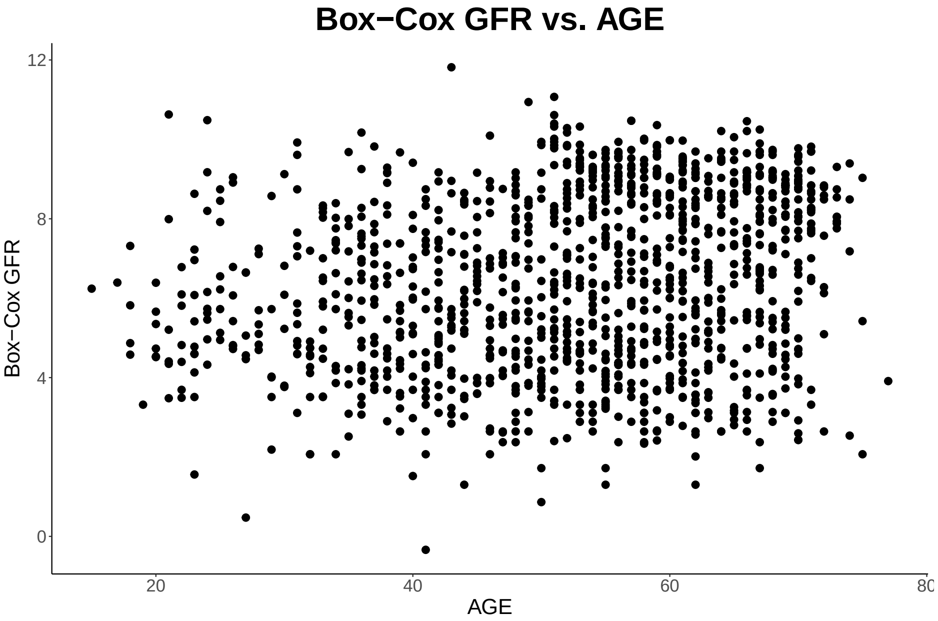


Figure 9: Point graph of Box-Cox GFR *vs.* AGE

Similar to *sbase* and *dbase*, AGE doesn't show obvious relationship with GFR. However, since the points in the graph above are uniformly distributed (no weird shape or evidence of heteroscedasticity), we don't consider transforming AGE.

The left predictors *Sex* and *black* are binary, thus we don't transform them.

Model Fitting

Now that all the variables are in the right format (although we may make modifications later due to further considerations), we start to fit a linear model now and execute variable selection. Furthermore, we consider throw interaction terms in the model and do variable selection on those terms too.

We first fit a full linear model with all the given predictors (transformed), including: log(*bascre*), *sbase*, *dbase*, log(*baseu*), *AGE*, *SEX* and *black*. The results are as follows:

Table 2: Results of linear regression including all predictors

	Estimate	Std. Error	Pr(>|t|)
(Intercept)	8.400	0.284	0.000
log(bascre)	-3.279	0.064	0.000
sbase	0.000	0.002	0.918
dbase	0.001	0.004	0.803
log(baseu)	-0.170	0.025	0.000
AGE	-0.007	0.003	0.013
SEX	0.661	0.065	0.000
black	0.189	0.111	0.089

Then, we applied stepAIC with backward selection to choose the optimal model. This procedure removes predictors that cannot make significant contributions to the model, which is measured by AIC (AIC is a criteria including numeric indicator of goodness-of-fit and the panelty for model complexity; the removal of a predictor will reduce the goodness-of-fit but simplify the model). After implementing this, two predictors *sbase* and *dbase* are removed from the model. We mentioned before that these two predictors may have significant linear relationship and can lead to collinearity. However, now that both of them are removed, we will not discuss this concern. The optimal model is as follows:

Table 3: Results of linear models after AIC selection

	Estimate	Std. Error	Pr(>|t|)
(Intercept)	8.510	0.150	0.000
log(bascre)	-3.281	0.064	0.000
log(baseu)	-0.170	0.025	0.000
AGE	-0.007	0.003	0.009
SEX	0.660	0.064	0.000
black	0.182	0.109	0.095

Now that we've selected the predictors, we consider potential interaction terms in this model. To do this, we apply the same strategy as we fit the previous model: we throw all possible interaction terms in this model, and then use AIC backward selection to find the final model.

Table 4: Results of linear models including interaction terms after AIC selection

	Estimate	Std. Error	Pr(>|t|)
(Intercept)	8.253	0.317	0.000
log(bascre)	-3.419	0.292	0.000
log(baseu)	-0.473	0.110	0.000
SEX	1.659	0.297	0.000
AGE	-0.007	0.006	0.264
black	-0.533	0.267	0.046
log(bascre):log(baseu)	0.265	0.047	0.000
log(bascre):SEX	-0.470	0.117	0.000
log(bascre):AGE	0.009	0.005	0.113
log(bascre):black	0.513	0.236	0.030
log(baseu):AGE	0.003	0.002	0.159
SEX:AGE	-0.014	0.005	0.006
SEX:black	0.535	0.218	0.014

The AIC selection procedure chooses seven interaction terms in our model: $\log(bascre) \times \log(baseu)$, $\log(bascre) \times SEX$, $\log(bascre) \times AGE$, $\log(bacres) \times black$, $\log(baseu) \times AGE$, $SEX \times AGE$, $SEX \times black$, see the table above.

Table 5: Comparison of model with and without interaction terms

	Adjusted R squared	AIC	BIC
Model without interaction terms	0.763	3731.973	3767.884
Model with interaction terms	0.771	3695.761	3767.583

We can see that the model with interaction terms have better performance in terms of adjusted R^2, AIC and BIC, all of which are measurements of goodness-of-fit with model complexity trade-off, ie. number of parameters. Thus, we choose the model with interaction terms to be our final candidate.

Model Diagnosis

We now examine the fit of our model to make further modifications to improve it. More specifically, we plot fitted values *vs.* residuals to check constancy of variance; we plot residuals *vs.* predictors to check linearity; we plot Q-Q plot to check normality; we identified potential outliers and influence points according to Cook's distance. We make changes to our model accordingly during the examination.

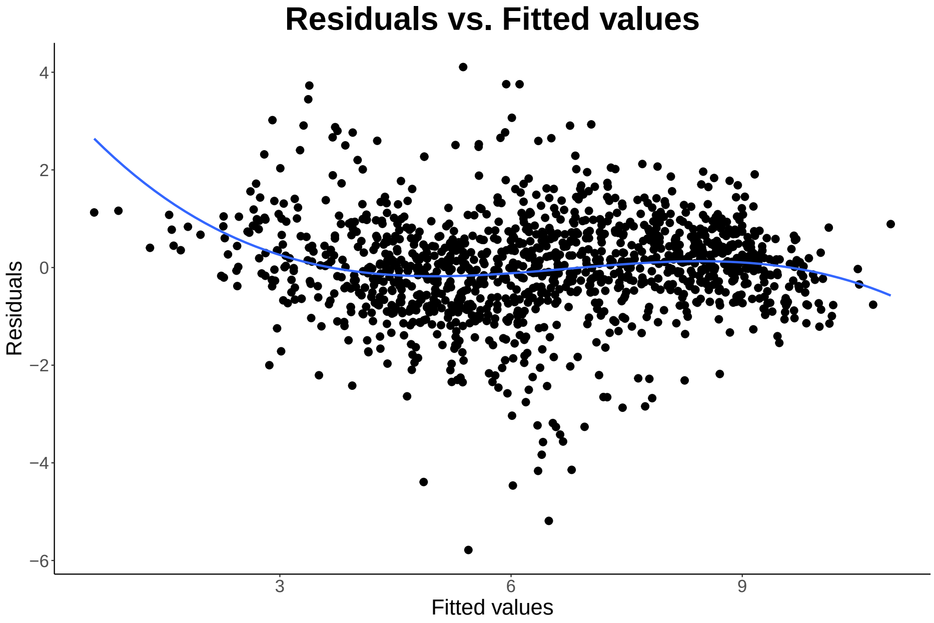


Figure 10: Residuals *vs.* Fitted values

According to the figure above, the residuals are almost evenly distributed around 0. The ranges of residuals are slightly wider when fitted values are small than when fitted values are large, and the range is the widest when fitted values are around 6, which, however, is probably caused by a few potential outliers. There seems to be a $y = a_0 + a_1x + a_2x^2 + a_3x^3$ function-shape, but in general these points are uniformly distributed. Since the range of residuals pretty much stay the same, we consider there is no evidence of heteroskedasticity.

Now we plot residuals *vs.* predictors to check linearity, that is, if the predictors should take another form (ie. quadratic). Note that we only look at continuous predictors because we cannot do much transformation to categorical predictors.

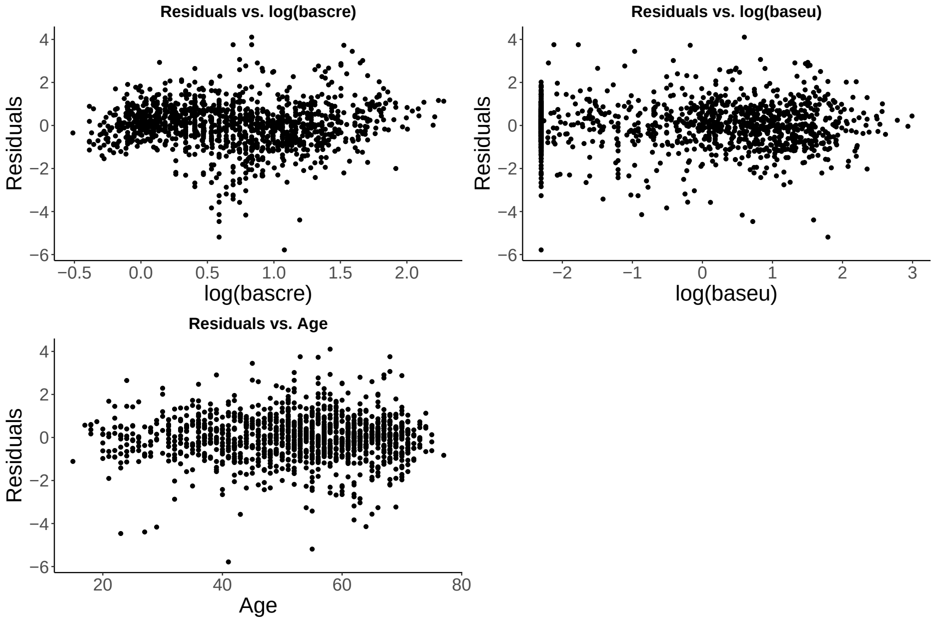


Figure 11: Residuals *vs.* Predictors

As shown by Figure 11, there are no wierd shapes in none of the plots and the points in the three figures are almost uniformly distributed. Thus, we justify the linearity and don't need to transform the predictors anymore.

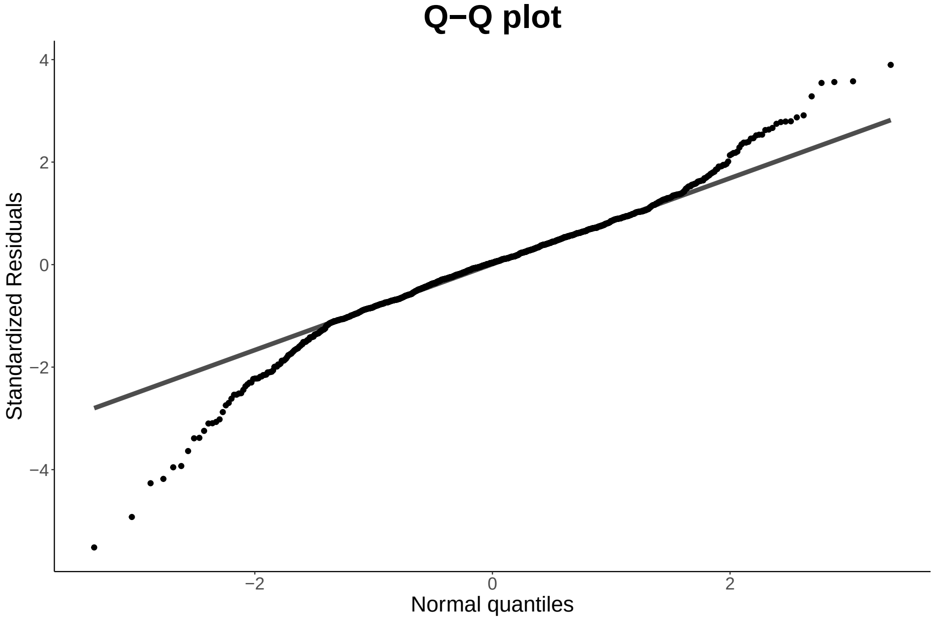


Figure 12: Q-Q plot

According to figure above, the standardized residuals do not exactly fit a normal distribution. However, since there is a nice linearity in the middle and the straight line only breaks at two sides, we consider it to be approximately fitting a normal distribution.

Now, we check influential points in our model.

Firstly, we list all outliers we detect using studentized residuals. Generally speaking, studentized residual for the ith observation is the standardized residual calculated based on the model fitted without the ith observation. It's more sensitive to observations who have outlying outcome value. If one observation's studentized residual is too large, then it basically means our model doesn't apply to this observation, thus including this observation may hurt the model-fitting.

Table 6: Outliers based on studentized residuals

	Studentized Residual	Bonferroni p
558	-5.585	0.000
92	-4.971	0.001
195	-4.295	0.023
189	-4.209	0.034

However, even though outliers are questionable. However, they do not always have significant impact on the model and points which have significant impact on the model may not have extreme studentized residuals. Now, we look for influential points, the inclusion of which in the model may hurt the model-fitting. We recognize those points by Cook's distance and DFFITS (difference in fitted values standardized). Cook's distance and DFFITS both measure the difference in fitted values for models with and without the ith observation that are standardized by some quantity. The rule of thumb is: observations with Cook's distance larger than $\frac{4}{n-p-1}$ or DFFITS larger than $2\sqrt{(p+1)/n}$ are considered to be influential points, where n is sample size and p is the number of predictors (including interaction terms).

Table 7: Influential points

Observation index	Cook's distance	DFFITS
10	0.003	0.209
11	0.010	0.362
16	0.004	0.225
17	0.007	0.299
20	0.006	0.285
25	0.005	0.254
34	0.004	0.221
35	0.016	0.458
47	0.003	0.209
49	0.003	0.210
50	0.011	0.384
53	0.008	0.325
58	0.007	0.305
59	0.010	0.368
67	0.003	0.207
84	0.004	0.223
85	0.004	0.218
90	0.003	0.206
108	0.004	0.214
152	0.004	0.231
239	0.017	0.464
241	0.008	0.316
286	0.006	0.277
324	0.006	0.279
340	0.011	0.373
345	0.007	0.307
363	0.012	0.403
400	0.004	0.237
1156	0.011	0.376
1161	0.006	0.289
1199	0.009	0.340
1239	0.005	0.265
1245	0.006	0.283

There are 65 observations identified as influential points by Cook's distance and 34 observations by DFFITS. Finally, there are 33 observations identified by both criteria. How to deal with these influential points, on the other hand, is another problem. We can examine the source of these points to see whether there are records error; or we consider expand our model by including more parameters; or we can totally abandon these observations. Each method has their merits and disadvantages, and should be carried on with reasonable assumptions. For now, however, due to the limit of time and relatively small number of influential points, we don't take further actions and just keep them in the model.

Predictive Plots

Finally, we enclose this report with some predictive plots, where we vary a few predictors while hold the others constant (i.e the mean in the data set) and see how fitted values change.

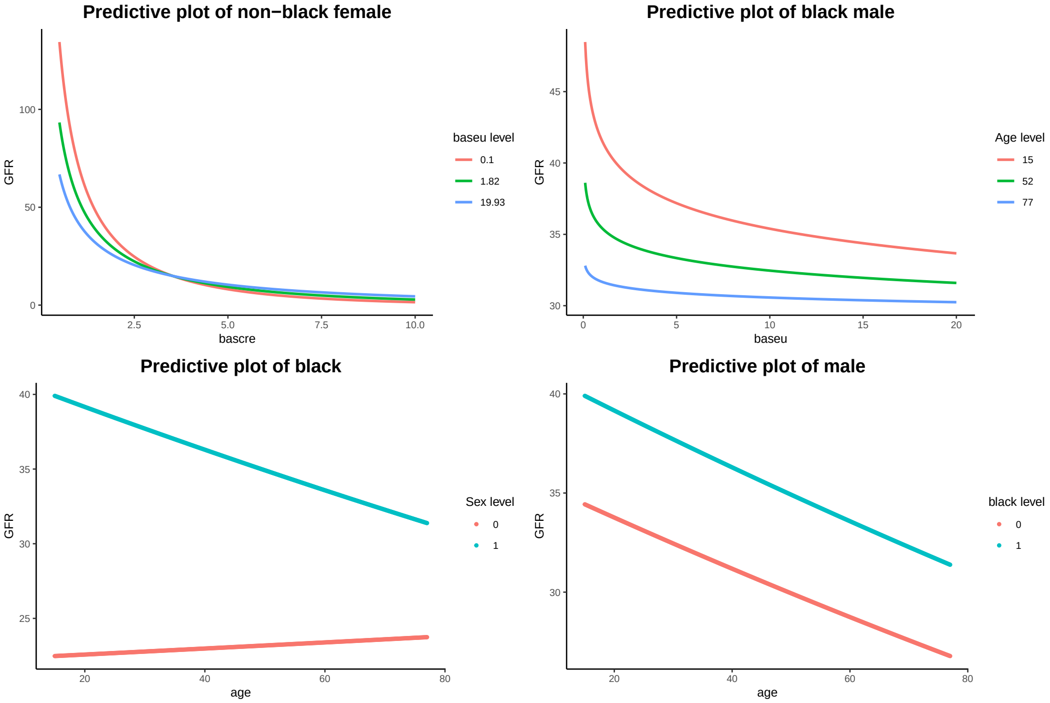


Figure 13: Predictive Plots

For plot on the upper left, GFR decreases as bascre increases and GFR of people with higher baseu level decreases faster; for plot on the upper right, GFR decreases as baseu increases and GFR of older people decreases faster; for plot on the bottom left, GFR of black female increases as age increases, while GFR of black male decreases as age increases; for plot on the bottom right, GFR of both black male and non-black male decrease as they grow older and black male have higher level of GFR compared to non-black male (of the same age).

Appendix for Code

```r
knitr::opts_chunk$set(dev = 'pdf', echo = FALSE, out.width = 100, warning = FALSE,
                      message = FALSE,fig.width=12, fig.height=8)
#Install packages
library(dplyr)
library(stringr)
library(tidyr)
library(broom)
library(rlist)
library(knitr)
library(kableExtra)
library(ggplot2)
library(rmarkdown)
library(stargazer)
library(float)
library(MASS)
library(EnvStats)
library(qqplotr)
library(ggpubr)
library(ggrepel)
library(car)
#read data
baseseg = read.csv('baseseg.csv')
predictors = c('bascre', 'sbase', 'dbase', 'baseu', 'AGE', 'SEX', 'black')
#construct summary table
data = baseseg%>%dplyr::select(predictors, 'gfr', 'X')
n_col = ncol(data)
#summary function
summ_func = function(x){
  return(c(min(x, na.rm = TRUE), quantile(x, 0.25, na.rm = TRUE), median(x, na.rm = TRUE), mean(x, na.rm
}
structure = round(apply(data[,-n_col], 2, summ_func),2)
rownames(structure) = c('Min', '1st Quantile', 'Median', 'Mean', '3rd Quantile', 'Max', "Percent of Mis
kable(structure, caption = 'Summary of variables', align = 'c', booktabs = TRUE, linesep = '')%>%kable_
#get rid of missing data
data_use = na.omit(data)
#plot histogram
ggplot(data_use, aes(x = gfr)) + geom_histogram(aes(y = ..count..)) + geom_density(aes(y = ..count.. *6
  ggtitle('Histogram of GFR') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
library(MASS)
#histogram after box cox transformation
bc = MASS::boxcox(lm(gfr~1, data = data_use), plotit = FALSE)
opt_lambda = bc$x[which(bc$y==max(bc$y))]
data_use$gfr = boxcoxTransform(data_use$gfr, lambda = opt_lambda)
ggplot(data_use, aes(x = gfr)) + geom_histogram(aes(y = ..count..)) + geom_density(aes(y = ..count../2)
  ggtitle('Histogram of Box-Cox GFR') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
#point plot
ggplot(data_use) + geom_point(aes(x = bascre, y = gfr), size = 3) + theme_classic() + labs(x = 'Serum C
  ggtitle('Box-Cox GFR vs. bascre') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
```

```
ggplot(data_use) + geom_point(aes(x = log(bascre), y = gfr), size = 3) + theme_classic() + labs(x = 'log
  ggtitle('Box-Cox GFR vs. log(bascre)') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
ggplot(data_use) + geom_point(aes(x = sbase, y = gfr), size = 3) + theme_classic() + labs(x = 'Systolic
  ggtitle('Box-Cox GFR vs. sbase') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
  ggplot(data_use) + geom_point(aes(x = dbase, y = gfr), size = 3) + theme_classic() + labs(x = 'Diastol
  ggtitle('Box-Cox GFR vs. dbase') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
  ggplot(data_use) + geom_point(aes(x = baseu, y = gfr), size = 3) + theme_classic() + labs(x = 'Urine
  ggtitle('Box-Cox GFR vs. baseu') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
  ggplot(data_use) + geom_point(aes(x = log(baseu), y = gfr), size = 3) + theme_classic() + labs(x = 'l
  ggtitle('Box-Cox GFR vs. log(baseu)') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
  ggplot(data_use) + geom_point(aes(x = AGE, y = gfr), size = 3) + theme_classic() + labs(x = 'AGE', y
  ggtitle('Box-Cox GFR vs. AGE') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
#fit model with all predictors
lm.full = lm(gfr ~ log(bascre) + sbase + dbase + log(baseu) + AGE + SEX + black, data = data_use)
#summary(lm.full)
coef = round(as.data.frame(summary(lm.full)$coefficients[,c(1,2,4)]),3)
kable(coef, caption = 'Results of linear regression including all predictors', align = 'c', booktabs =
#backward AIC selection
back = stepAIC(lm.full, direction = 'backward', trace = 0)
#summary(back)
back_coef = round(as.data.frame(summary(back)$coefficients[,c(1,2,4)]),3)
kable(back_coef, caption = 'Results of linear models after AIC selection', align = 'c', booktabs = TRUE
#include all possible interaction terms
lm.inter = lm(gfr ~ log(bascre) + log(baseu) + SEX + AGE + black + log(bascre)*log(baseu) + log(bascre)
#summary(lm.inter)
#do backward AIC selection on interaction terms
back.inter = stepAIC(lm.inter, direction = 'backward', trace = 0)
back.inter.coef = round(as.data.frame(summary(back.inter)$coefficients[,c(1,2,4)]),3)
kable(back.inter.coef, caption = 'Results of linear models including interaction terms after AIC selecti
#summary(back.inter)
#compare models with criterai such AIC
lm.nointe.cri = c(summary(back)$adj.r.squared, AIC(back), BIC(back))
lm.inte.cri = c(summary(back.inter)$adj.r.squared, AIC(back.inter), BIC(back.inter))

#make a table
criteria = round(as.data.frame(rbind(lm.nointe.cri, lm.inte.cri)),3)
rownames(criteria) = c('Model without interaction terms', 'Model with interaction terms')
colnames(criteria) = c('Adjusted R squared', 'AIC', 'BIC')
kable(criteria, caption = 'Comparison of model with and without interaction terms', align = 'c', bookta
#fitted values vs. residuals
fit.residual = data.frame(fit = fitted.values(back.inter), res = residuals(back.inter), stad.res = rstan
ggplot(fit.residual, aes(x = fit, y = res)) + geom_point(size = 3) + stat_smooth(method='lm', se = FALS
  ggtitle('Residuals vs. Fitted values') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
thresh = 4
#residuals vs. predictors
q.bascre = ggplot(fit.residual, aes(x = log(bascre), y = res)) + geom_point() + theme_classic() + labs(
  ggtitle('Residuals vs. log(bascre)') + theme(text = element_text(size=20), plot.title = element_text(h

q.baseu = ggplot(fit.residual, aes(x = log(baseu), y = res)) + geom_point() + theme_classic() + labs(x =
  ggtitle('Residuals vs. log(baseu)') + theme(text = element_text(size=20), plot.title = element_text(h
q.age = ggplot(fit.residual, aes(x = AGE, y = res)) + geom_point() + theme_classic() + labs(x = 'Age', y
  ggtitle('Residuals vs. Age') + theme(text = element_text(size=20), plot.title = element_text(hjust = 0

ggarrange(q.bascre, q.baseu, q.age)

#qq plot
ggplot(fit.residual, aes(sample = stad.res)) + stat_qq_line(size = 2) + stat_qq_point(size = 2)+theme_cl
  ggtitle('Q-Q plot') +
  theme(text = element_text(size=20), plot.title = element_text(hjust = 0.5, size = 30, face = 'bold'))
#calculate relevant quantities to identify influential points and outliers
influ = influence.measures(back.inter)

rst = rstudent(back.inter)

cook = cooks.distance(back.inter)

dff = dffits(back.inter)

outlier = outlierTest(back.inter)
#outlier table
outlier.table = round(cbind(outlier[[1]],  outlier[[3]]),3)
colnames(outlier.table) = c('Studentized Residual', 'Bonferroni p')
kable(outlier.table, caption = 'Outliers based on studentized residuals', align = 'c', booktabs = TRUE,
#influential points table
n = dim(data_use)[1]
p = length(back.inter$coefficients) - 1

inf_by_cook = which(cook>(4/(n-p-1)))
inf_by_dff = which(dff>2*sqrt((p+1)/n))

influ = base::intersect(inf_by_cook, inf_by_dff)

inf.table = round(cbind(influ, cook[influ], dff[influ]),3)
colnames(inf.table) = c('Observation index', "Cook's distance", 'DFFITS')
rownames(inf.table) = NULL

kable(inf.table, caption = 'Influential points', align = 'c', booktabs = TRUE, linesep = '')%>%kable_sty
#make predictive plots
bascre.data = data_use$bascre
baseu.data = data_use$baseu
age.data = data_use$AGE
#summary(bascre.data)
#summary(baseu.data)
#summary(age.data)

#generate fake data
bascre.ge = seq(0.5, 10, length.out = 500)
baseu.ge = seq(0.1, 20, length.out = 500)
age.ge = seq(15, 77, length.out = 500)
```

```
fake.data.1 = data.frame(AGE = round(mean(age.data),0), SEX = 0, black = 0, bascre = bascre.ge, baseu =
fake.data.2 = data.frame(AGE = round(mean(age.data),0), SEX = 0, black = 0, bascre = bascre.ge, baseu =
fake.data.3 = data.frame(AGE = round(mean(age.data),0), SEX = 0, black = 0, bascre = bascre.ge, baseu =

fake.data_1 = rbind(fake.data.1, fake.data.2, fake.data.3)
y_1 = predict(back.inter, fake.data_1)
fake.data_1$gfr = (opt_lambda*y_1 +1)^(1/opt_lambda)

#bascre vs. gfr
p1 = ggplot(data = fake.data_1) + geom_line(aes(x = bascre, y = gfr, color = as.character(baseu)), size
  ggtitle('Predictive plot of non-black female') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 15, face = 'bold'))

fake.data.21 = data.frame(AGE = round(mean(age.data),0), SEX = 1, black = 1, bascre = mean(bascre.data)
fake.data.22 = data.frame(AGE = round(min(age.data),0), SEX = 1, black = 1, bascre = mean(bascre.data),
fake.data.23 = data.frame(AGE = round(max(age.data),0), SEX = 1, black = 1, bascre = mean(bascre.data),

fake.data_2 = rbind(fake.data.21, fake.data.22, fake.data.23)
y_2 = predict(back.inter, fake.data_2)
fake.data_2$gfr = (opt_lambda*y_2 +1)^(1/opt_lambda)

#baseu vs. gfr
p2 = ggplot(data = fake.data_2) + geom_line(aes(x = baseu, y = gfr, color = as.character(AGE)), size =
  ggtitle('Predictive plot of black male') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 15, face = 'bold'))

fake.data.31 = data.frame(AGE = age.ge, SEX = 0, black = 1, bascre = mean(bascre.data), baseu = mean(ba
fake.data.32 = data.frame(AGE = age.ge, SEX = 1, black = 1, bascre = mean(bascre.data), baseu = mean(ba
fake.data.41 = data.frame(AGE = age.ge, SEX = 1, black = 0, bascre = mean(bascre.data), baseu = mean(ba
fake.data.42 = data.frame(AGE = age.ge, SEX = 1, black = 1, bascre = mean(bascre.data), baseu = mean(ba

fake.data_3 = rbind(fake.data.31, fake.data.32)
y_3 = predict(back.inter, fake.data_3)
fake.data_3$gfr = (opt_lambda*y_3 +1)^(1/opt_lambda)

fake.data_4 = rbind(fake.data.41, fake.data.42)
y_4 = predict(back.inter, fake.data_4)
fake.data_4$gfr =(opt_lambda*y_4 +1)^(1/opt_lambda)

#age vs. gfr with sex levels
p3 = ggplot(data = fake.data_3) + geom_point(aes(x = AGE, y = gfr, color = as.character(SEX)), size = 1
  ggtitle('Predictive plot of black') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 15, face = 'bold'))
#age vs. gfr with black levels
p4 = ggplot(data = fake.data_4) + geom_point(aes(x = AGE, y = gfr, color = as.character(black)), size =
  ggtitle('Predictive plot of male') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 15, face = 'bold'))

ggarrange(p1,p2,p3,p4)
```

HW2 (PART B)

Yimo Zhang

Fall 2019

```
library(MASS)
library(ggplot2)
library(ggpubr)
```

```
Loading required package: magrittr
```

a

```
set.seed(123)
n_sim = 1000
y = rnorm(100, 10, 2)
x = rnorm(100, 3, 1)
```

b

```
lm.1 = lm(y~x)
p.1 = summary(lm.1)$coefficients[-1,4]
```

c

```
p.1
```

```
[1] 0.6245623
```

The $p-value$ is 0.625, which is way larger than the 0.05 threshold. Since x and y are indepednent (thus not linear relationship) from the way to generate them, it makes sense that the $p-value$ is this big.

d

```
set.seed(123)
sim1 = function(n_sim){
  p.list = c()
  for(i in 1:n_sim){
    y = rnorm(100, 10, 2)
    x = rnorm(100, 3, 1)
    lm.1 = lm(y~x)
    p = summary(lm.1)$coefficients[2,4]
    p.list = c(p.list, p)
  }
  return(p.list)
}
p.s = sim1(1000)
```

e

```
mean(p.s<0.05)
```

```
[1] 0.053
```

The proportion of times when the $p-value < 0.05$ is 0.053. This matches my intuition because x and y are independent from the way we generate them, thus there should be no linear relationship between them, that is to say, if there is any linear relationship, it's purely happening by chance, and this is why the proportion of times is so small.

The type of error depends on what we believe is null hypothesis. Assume that the null hypothesis is x and y are independent, or say, the slope is not related to the outcome, (we generate data from null hypothesis), then we are calculating Type I error, which is the probability of rejecting null hypothesis given null hypothesis.

f

```
set.seed(123)
x.2 = rnorm(100, 3, 1)
y.2 = rnorm(100, 10+x.2, 1)

lm.2 = lm(y.2~x.2)
p.2 = summary(lm.2)$coefficients[2,4]
p.2
```

```
[1] 3.497142e-14
```

The $p-value$ is very small (near 0), which makes sense because $E[Y|X] = 10 + X$ from the way we generate the data. Thus the linear relationship between X and Y should be significant.

```
sim2 = function(n_sim){
  p.list = c()
  for(i in 1:n_sim){
    x.2 = rnorm(100,3,1)
    y.2 = rnorm(100,10+x.2, 1)
    lm.2 = lm(y.2~x.2)
    p = summary(lm.2)$coefficients[2,4]
    p.list = c(p.list, p)
  }
  return(p.list)
}

p.2.list = sim2(1000)
mean(p.2.list<0.05)
```

```
[1] 1
```

Different from the first simulation, the proportion of times when p_values are less than 0.05 is 100%, which means coefficients from all linear regression are significant. The probability-related interpretation of this proportion is: given that X and Y are linearly dependent by $E[Y|X] = 10 + X$, the probability that Y and X are actually linearly dependent is 1, which is the power, and also, $1 -$ type II error.

2.

a

```
set.seed(123)
Y = rnorm(100, 10, 2)
X = mvrnorm(100, c(1,2,3), diag(3))
```

b

```
lm.3 = lm(Y~X)
p.values = summary(lm.3)$coefficients[2:4, 4]
p.values
```

```
       X1        X2        X3 
0.6362654 0.2017467 0.6684060 
```

c

```
set.seed(123)
sim2.5 = function(n_sim){
  p.list = c()
    for(i in 1:n_sim){
      Y = rnorm(100, 10, 2)
      X = mvrnorm(100, c(1,2,3), diag(3))
      lm.3 = lm(Y~X)
      p.values = summary(lm.3)$coefficients[2:4, 4]
      p.list = rbind(p.list, p.values)
    }
  return(p.list)
}

p.list = sim2.5(1000)
apply(p.list, 2, function(x) sum(x<0.05))
```

```
X1 X2 X3
43 47 50
```

For X_1, 43 out of 1000 p-values are significant at 0.05 level; 47 for X_2 and 50 for X_3.

d

```
p.min.list = apply(p.list, 1, min)
sum(p.min.list<0.05)
```

```
[1] 132
```

132 out of 1000 minimum p_values are less than 0.05, thus significant under that level. For better interpretation, we write down this model as:

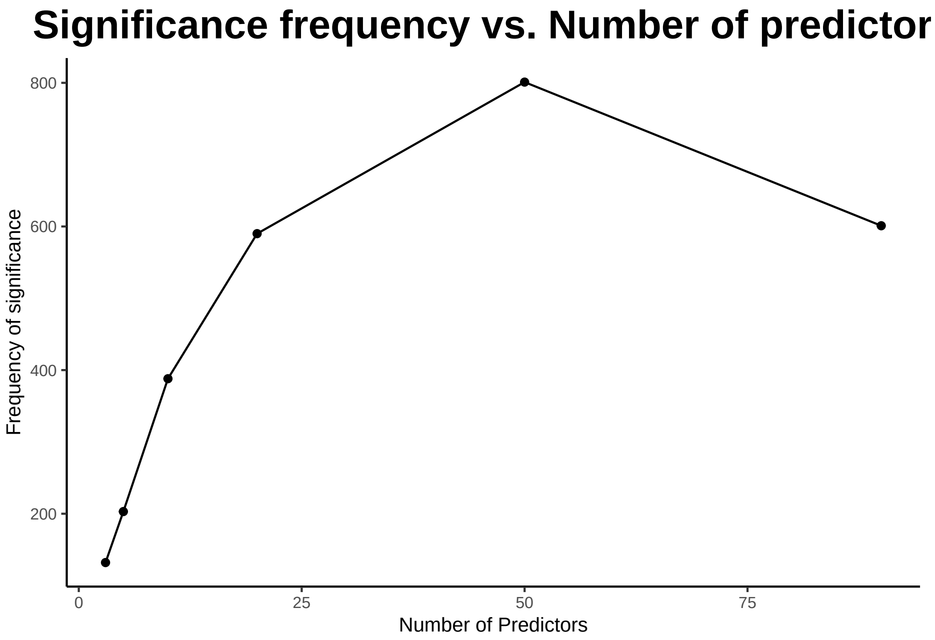
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$$

. This result is different from the question above from two aspects: 1) $132 > \max\{43, 47, 50\}$, because when one of the coefficients, say β_1 is non-significant at 0.05 level, the other coefficients may be significant; 2) $132 < 43 + 47 + 50 = 140$, because there are cases when more than one coefficients are significant.

e

```
P = c(3, 5, 10, 20, 50, 90)
sim4 = function(P, n_sim){
  sig.times = c()
  for(p in P){
    p.list = c()
    for(i in 1:n_sim){
      Y = rnorm(100, 10, 4)
      X = mvrnorm(100, seq(p), diag(p))
      model = lm(Y~X)
      p.values = summary(model)$coefficients[-1, 4]
      p.list = c(p.list, min(p.values))
    }
  sig.times = c(sig.times, sum(p.list<0.05))
  }
  return(sig.times)
}
set.seed(123)
```

```
sig = sim4(P, 1000)
sig.table = data.frame(p = P, sig = sig)
ggplot(sig.table) + geom_point(aes(x = p, y = sig)) + geom_line(aes(x = p, y = sig)) + theme_classic()
  ggtitle('Significance frequency vs. Number of predictors') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 20, face = 'bold'))
```



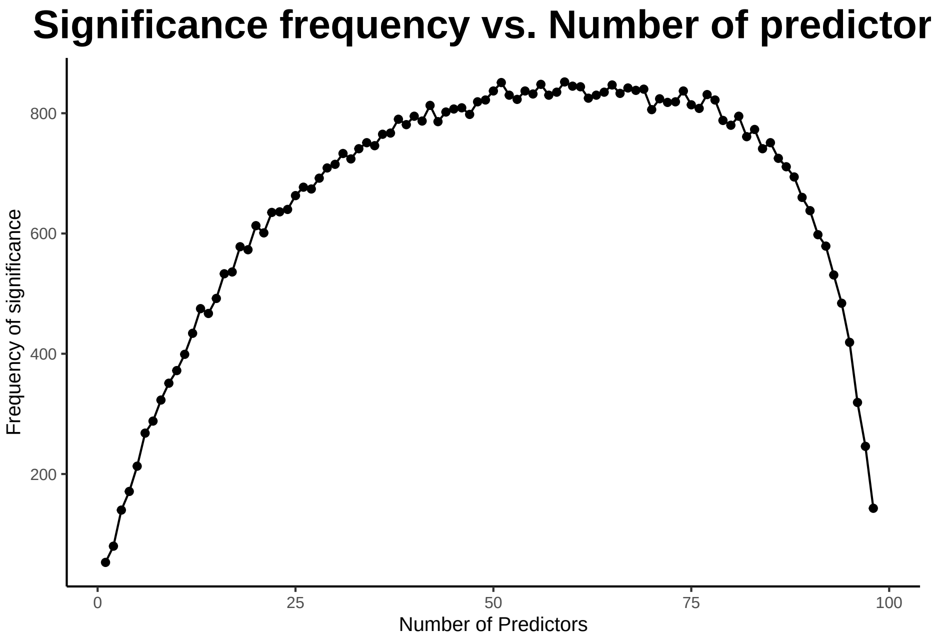
The number of the minimum p_values less than 0.05 first increases as the number of predictors increases; however, after reaching the peak at $p = 50$, the number of minimum p_values less than 0.05 decreases. From the way we generate the data, X and Y are independent; however, as we add more predictors to X, the chance that Y is calculated as linearly dependent on X increases, although this indepence is happening by chance (that is, X and Y just look like being dependent, but they are not according to their sources). Moreover, as we keep adding more predictors to X, the number of predictors is approaching the number of observations, which generates unreliable coefficient estimates which have variance, and the results become non-significant. This is one of the consequences of overfitting – leading to estimates with large variance.

f

```
P = seq(99)
set.seed(123)
sig.sim = sim4(P, 1000)
sig.table = data.frame(p = P, sig = sig.sim)
ggplot(sig.table) + geom_point(aes(x = p, y = sig)) + geom_line(aes(x = p, y = sig)) + theme_classic()
  ggtitle('Significance frequency vs. Number of predictors') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 20, face = 'bold'))
```

```
Warning: Removed 1 rows containing missing values (geom_point).
```

```
Warning: Removed 1 rows containing missing values (geom_path).
```



The result of this simulation is consistent with the previous one. We can see that the significance frequency reaches its peak when the number of predictors is between 50 and 75, and starts to decrease rapidly when the number of predictors approaching 100.

3.

```
sim5 = function(n_sim){
  p.list = c()
    for(i in 1:n_sim){
      Y = rnorm(100, 10, 2)
      X = mvrnorm(100, c(1,2,3), 0.5+0.5*diag(3))
      lm.3 = lm(Y~X)
      p.values = summary(lm.3)$coefficients[2:4, 4]
      p.list = rbind(p.list, p.values)
    }
  return(p.list)
}
set.seed(123)
times = sim5(1000)
apply(times, 2, function(x) sum(x<0.05))
```

```
X1 X2 X3
47 47 46
```

47 out of 1000 p_values of X_1 are less than 0.05; 47 for X_2 and 46 for X_3.

```
p.min.list = apply(times, 1, min)
sum(p.min.list<0.05)
```

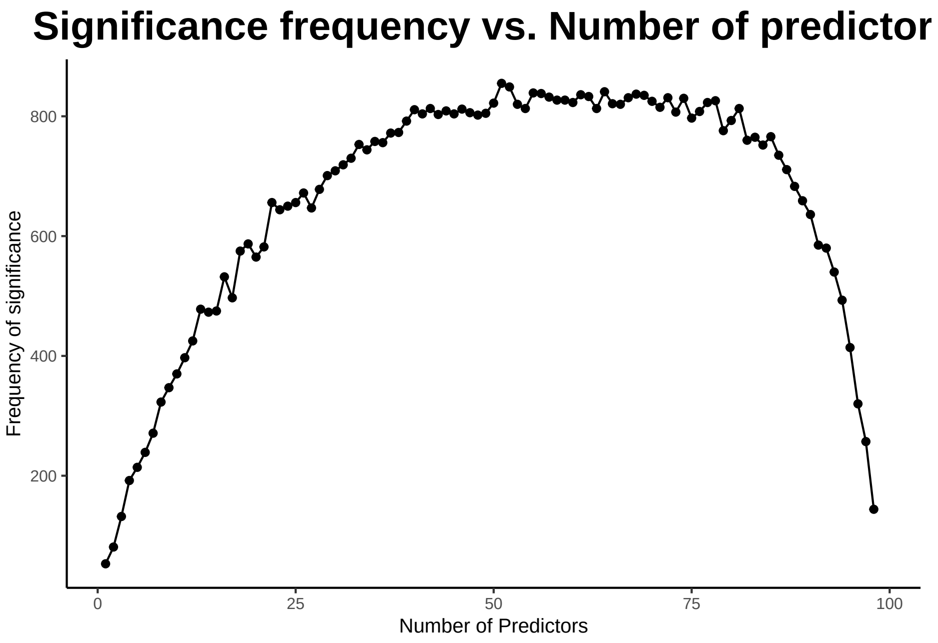
```
[1] 125
```

125 out of 1000 minimum p_values are less than 0.05, thus significant.

```r
P = seq(99)
sim7 = function(P, n_sim){
  sig.times = c()
  for(p in P){
    p.list = c()
    for(i in 1:n_sim){
      Y = rnorm(100, 10, 4)
      X = mvrnorm(100, seq(p), 0.5+0.5*diag(p))
      model = lm(Y~X)
      p.values = summary(model)$coefficients[-1, 4]
      p.list = c(p.list, min(p.values))
    }
  sig.times = c(sig.times, sum(p.list<0.05))
  }
  return(sig.times)
}
set.seed(123)
sig = sim7(P, 1000)
sig.table = data.frame(p = P, sig = sig)
ggplot(sig.table) + geom_point(aes(x = p, y = sig)) + geom_line(aes(x = p, y = sig)) + theme_classic() +
  ggtitle('Significance frequency vs. Number of predictors') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 20, face = 'bold'))
```

```
Warning: Removed 1 rows containing missing values (geom_point).
```

```
Warning: Removed 1 rows containing missing values (geom_path).
```



Frequency of significance (of minimum p_values) first increases as the number of predictors increases, and then decreases as the number predictors approaches the number of observations. This result is pretty much the same as the previous problem (similar plot), mostly because X and Y are independent in both cases from the way we generate them. Although the design matrix changes with each predictors being correlated to each other, this makes them neither entirely linearly dependent with each other (which will introduce collinearity) nor dependent with Y. Thus, the results are very similar.

4.

a

```
set.seed(123)


sim8 = function(n_sim, power, b, orig_size = 3){

  size = 3
  while(size<Inf){
     p.values = c()
  for(i in 1:n_sim){
    x = rnorm(size, 100, 1)
    y = rnorm(size, b*x, 1)
    model = lm(y~x)
    p = summary(model)$coefficients[-1,4]
    p.values = c(p.values, p)
  }
     if(mean(p.values<0.05) >= power){return(size)}
     size = size + 1
  }
}

size80 = sim8(100, 0.8, 1)
size80
```

```
[1] 13
```

In order to carry on this simulation: 1) when calculate the power, we use 100 simulations instead of 1000 in the previous questions, this is because running 1000 simulations take too much time, especially when the effect size is small; 2) the starting sample size is 3 because we have two parameters β_0, β_1 and the sample size should be larger than that to conduct valid analysis.

In order to reach a power of at least 80%, the sample size should be at least 13. # b

```
set.seed(123)
size90 = sim8(100, 0.9, 1)
size90
```

```
[1] 16
```

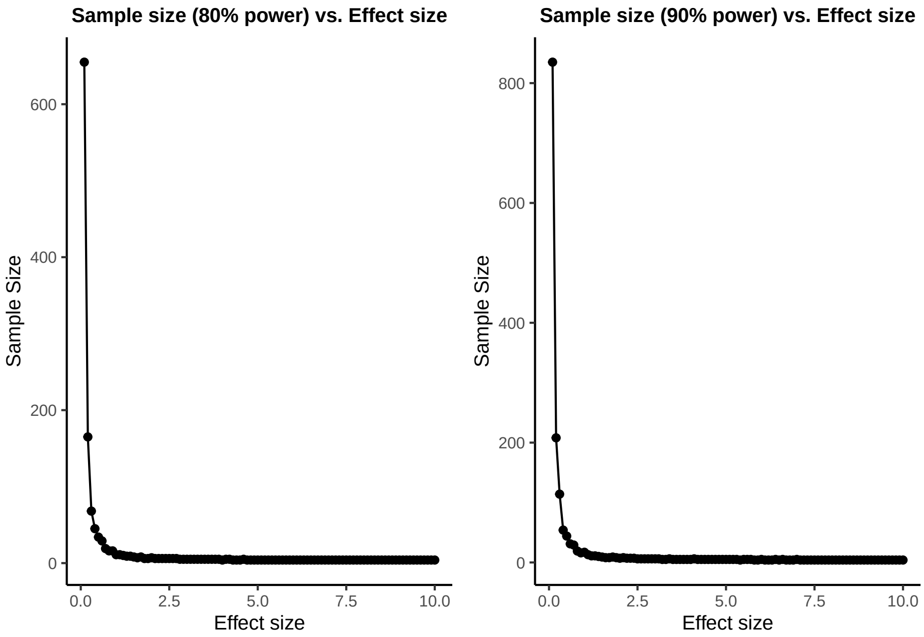
In order to reach a power of at least 90%, the sample size should be at least 16. # c

```
set.seed(123)
B = seq(0.1, 10, by = 0.1)
size.80 = c()
size.90 = c()

for(b in B){
  size.80 = c(size.80, sim8(100, 0.8, b))
  size.90 = c(size.90, sim8(100, 0.9, b))
}
table.power = data.frame(effect = B, size80 = size.80, size90 = size.90)
g80 = ggplot(table.power)  + geom_point(aes(x = effect, y = size80)) + geom_line(aes(x = effect, y = si
  ggtitle('Sample size (80% power) vs. Effect size') +
  theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 10, face = 'bold'))

g90 = ggplot(table.power)  + geom_point( aes(x = effect, y = size90)) + geom_line( aes(x = effect, y = s
theme(text = element_text(size=10), plot.title = element_text(hjust = 0.5, size = 10, face = 'bold'))

ggarrange(g80, g90)
```



The closer the effect size is to 0, the larger the sample size is needed to reach a 80% or 90% power. This is because when b is smaller, the ‘distance’ between null hypothesis and alternative hypothesis is smaller, and we need more data to distinguish between them.