R Lab 11. Multicollinearity and Ridge Regression

1. Measuring Multicollinearity

```
> attach(mtcars)
> names(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear" "carb"
```

 # R shows rather high correlations among variables

```
> cor(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
mpg	1.0000000	-0.8521620	-0.8475514	-0.7761684	0.68117191	-0.8676594	0.41868403	0.6640389	0.59983243	0.4802848	-0.55092507
cyl	-0.8521620	1.0000000	0.9020329	0.8324475	-0.69993811	0.7824958	-0.59124207	-0.8108118	-0.52260705	-0.4926866	0.52698829
disp	-0.8475514	0.9020329	1.0000000	0.7909486	-0.71021393	0.8879799	-0.43369788	-0.7104159	-0.59122704	-0.5555692	0.39497686
hp	-0.7761684	0.8324475	0.7909486	1.0000000	-0.44875912	0.6587479	-0.70822339	-0.7230967	-0.24320426	-0.1257043	0.74981247
drat	0.6811719	-0.6999381	-0.7102139	-0.4487591	1.00000000	-0.7124406	0.09120476	0.4402785	0.71271113	0.6996101	-0.09078980
wt	-0.8676594	0.7824958	0.8879799	0.6587479	-0.71244065	1.0000000	-0.17471588	-0.5549157	-0.69249526	-0.5832870	0.42760594
qsec	0.4186840	-0.5912421	-0.4336979	-0.7082234	0.09120476	-0.1747159	1.00000000	0.7445354	-0.22986086	-0.2126822	-0.65624923
vs	0.6640389	-0.8108118	-0.7104159	-0.7230967	0.44027846	-0.5549157	0.74453544	1.0000000	0.16834512	0.2060233	-0.56960714
am	0.5998324	-0.5226070	-0.5912270	-0.2432043	0.71271113	-0.6924953	-0.22986086	0.1683451	1.00000000	0.7940588	0.05753435
gear	0.4802848	-0.4926866	-0.5555692	-0.1257043	0.69961013	-0.5832870	-0.21268223	0.2060233	0.79405876	1.0000000	0.27407284
carb	-0.5509251	0.5269883	0.3949769	0.7498125	-0.09078980	0.4276059	-0.65624923	-0.5696071	0.05753435	0.2740728	1.00000000

 # VIF = variance inflation factors are used to measure multicollinearity.

```
> install.packages("car")
> library(car)

> reg = lm( mpg ~ ., data=mtcars )
> vif(reg)
```

cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
15.373833	21.620241	9.832037	3.374620	15.164887	7.527958	4.965873	4.648487	5.357452	7.908747

 # In presence of multiple categories and dummy variables, generalized VIF are used, adjusted for the degrees of freedom (number of dummies per a categorical variable)

```
> setwd("C:\\Users\\baron\\Documents\\Teach\\615 Regression\\Data")
> HOMES = read.csv("HOME_SALES.csv")
> attach(HOMES)
> reg = lm( SALES_PRICE ~ .-ID-STYLE+as.factor(STYLE), data=HOMES )
> vif(reg)
```

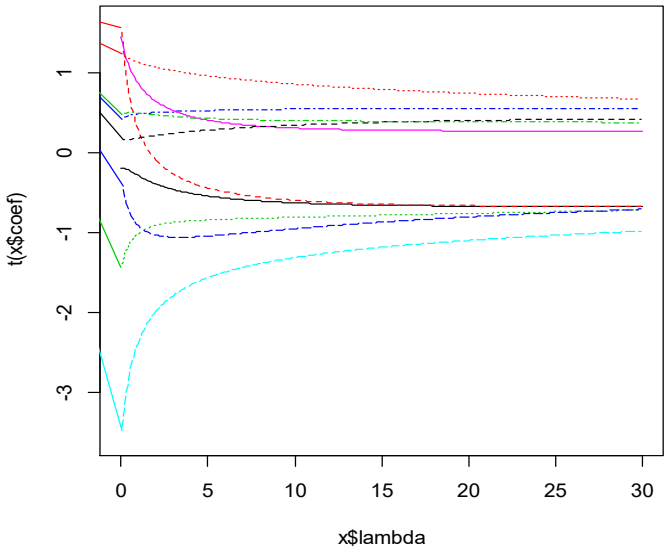
	GVIF	Df	GVIF^(1/(2*Df))
FINISHED_AREA	4.471114	1	2.114501
BEDROOMS	1.689177	1	1.299683
BATHROOMS	3.175930	1	1.782114
GARAGE_SIZE	1.659708	1	1.288296
YEAR_BUILT	1.941969	1	1.393545
LOT_SIZE	1.178829	1	1.085739
AIR_CONDITIONER	1.395114	1	1.181150
POOL	1.057359	1	1.028280
QUALITY	3.817176	2	1.397769
HIGHWAY	1.032433	1	1.016087
as.factor(STYLE)	2.185423	2	1.215861

2. Ridge Regression

 # You can run ridge regression with a given chosen lambda or try a whole range of lambdas.

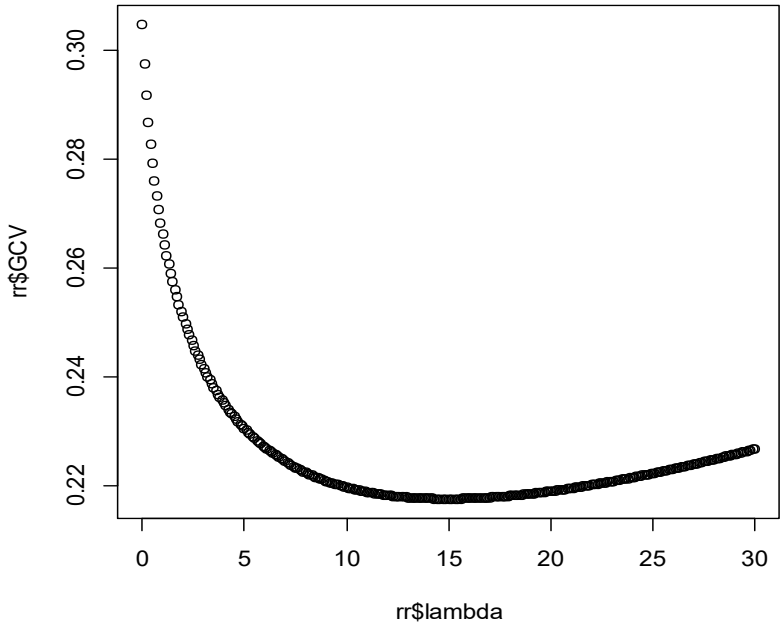
```
> install.packages("MASS")
> library(MASS)
> rr = lm.ridge( mpg ~ ., data=mtcars, lambda=0.5 )
> rr
                       cyl          disp            hp          drat            wt
14.853945860 -0.126513088  0.005661242 -0.016779725  0.886565240 -2.859953631
        qsec            vs            am          gear          carb
 0.607130107  0.341162437  2.396463996  0.693482725 -0.472630120

> rr = lm.ridge( mpg ~ ., data=mtcars, lambda=seq(0,30,0.1) )
> plot(rr)
```



 # These are ridge regression slopes as functions of λ
 # You can also look at the mean-squared prediction error obtained by cross-validation

```
> plot(rr$lambda, rr$GCV)
```



 # Which λ minimize MSPE? Here is how you can find it.

```
> select(rr)
modified HKB estimator is 2.58585
modified L-W estimator is 1.837435
smallest value of GCV  at 14.9

> rr.optimal = lm.ridge( mpg ~ ., data=mtcars, lambda=14.9 )
> coef(rr.optimal)
                  cyl         disp           hp         drat           wt
21.13638336 -0.37299085 -0.00527183 -0.01159978  1.05345707 -1.23062378
       qsec          vs          am         gear         carb
 0.16168503  0.77096188  1.62029355  0.54427196 -0.54685394
```

 # Prediction with ridge regression.
 # lm.ridge does not have a "predict" command…

```
> predict(rr.optimal, data=mtcars)
Error in UseMethod("predict") :
  no applicable method for 'predict' applied to an object of class "ridgelm"
```

 # But we know that ridge regression Is also a linear regression. So, we can compute predicted values "mpg.predicted" simply by the formula X*b, where X = design matrix and b = vector of slopes.
 # Here is an easy way to create matrix X .

```
> reg = lm( mpg ~ ., data=mtcars )
> X = model.matrix(reg)
> X
                  (Intercept) cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4                   1   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag               1   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710                  1   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive              1   6 258.0 110 3.08 3.215 19.44  1  0    3    1
   < truncated >

> mpg.predicted = X %*% coef(rr.optimal)
> plot(mpg, mpg.predicted)
```