Multiple Linear Regression: Credit Data

Prof. Kapitula

10/15/2020

This code is shared in ~/Sharedprojects/Kapitula/STA631/MLR

read: https://moderndive.com/6-multiple-regression.html read: Chapter 6 - 8 in Regression and Other Stories

Credit Data Analysis

For example, the Credit data set from ISLR records balance (average credit card debt for a number of individuals) as well as several quantitative predictors: age, cards (number of credit cards), education (years of education), income (in thousands of dollars), limit (credit limit), and rating (credit rating). In addition to these quantitative variables, we also have four qualitative variables: gender, student (student status), status (marital status), and ethnicity (Caucasian, African American or Asian).

```r
library(ISLR) #contains the credit data
library(tidyverse)
library(GGally)
library(moderndive)
library(skimr)
library(rstanarm)
library(bayesplot)
theme_set(theme_classic())
```

Below we read in the Credit data from the ISLR package for analysis. I rescale Limit and Balance to be in 1000s of dollars to make plots look a little neater.

```r
data(Credit, package = "ISLR")
Credit <- as_tibble(Credit)
Credit <- Credit %>%
  mutate(Limit1000=Limit/1000, Balance1000=Balance/1000)
#glimpse(Credit)
```

Check out 5 random cases.

```r
Credit %>% sample_n(size = 5)
```

```
## # A tibble: 5 x 14
##      ID Income Limit Rating Cards   Age Education Gender Student Married
##   <int>  <dbl> <int>  <int> <int> <int>     <int> <fct>  <fct>   <fct>
## 1   367   61.1  7871    564     3    56        14 " Mal~ No      Yes
## 2   156   19.5  1362    143     4    34         9 "Fema~ No      Yes
## 3   382  102.   8029    574     2    84        11 " Mal~ No      Yes
## 4   351   30.0  1561    155     4    70        13 "Fema~ No      Yes
## 5   186   30.4  4442    316     1    30        14 "Fema~ No      No
## # ... with 4 more variables: Ethnicity <fct>, Balance <int>, Limit1000 <dbl>,
## #   Balance1000 <dbl>
```

```r
#Credit %>% skim()
Credit %>% skim_without_charts()
```

Table 1: Data summary

Name	Piped data
Number of rows	400
Number of columns	14
Column type frequency:	
factor	4
numeric	10
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
Gender	0	1	FALSE	2	Fem: 207, Ma: 193
Student	0	1	FALSE	2	No: 360, Yes: 40
Married	0	1	FALSE	2	Yes: 245, No: 155
Ethnicity	0	1	FALSE	3	Cau: 199, Asi: 102, Afr: 99

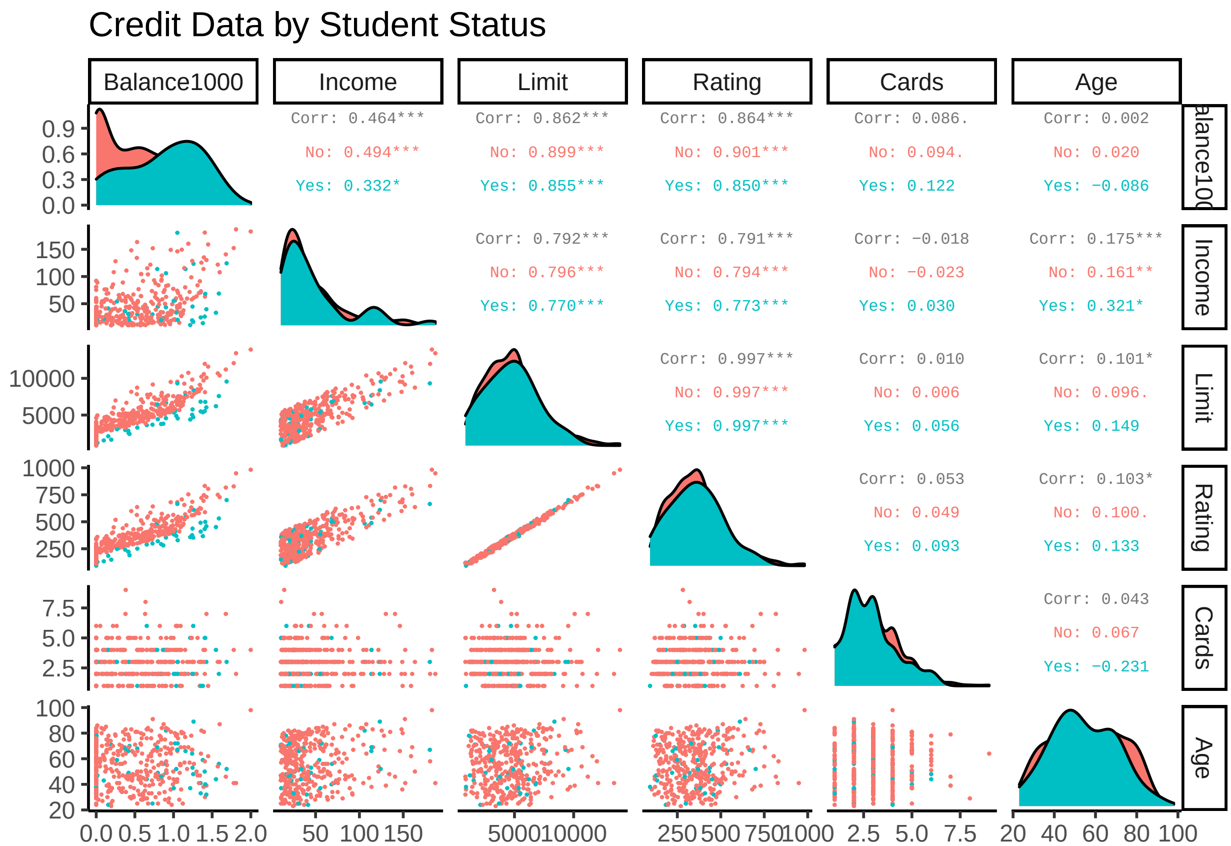
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
ID	0	1	200.50	115.61	1.00	100.75	200.50	300.25	400.00
Income	0	1	45.22	35.24	10.35	21.01	33.12	57.47	186.63
Limit	0	1	4735.60	2308.20	855.00	3088.00	4622.50	5872.75	13913.00
Rating	0	1	354.94	154.72	93.00	247.25	344.00	437.25	982.00
Cards	0	1	2.96	1.37	1.00	2.00	3.00	4.00	9.00
Age	0	1	55.67	17.25	23.00	41.75	56.00	70.00	98.00
Education	0	1	13.45	3.13	5.00	11.00	14.00	16.00	20.00
Balance	0	1	520.02	459.76	0.00	68.75	459.50	863.00	1999.00
Limit1000	0	1	4.74	2.31	0.86	3.09	4.62	5.87	13.91
Balance1000	0	1	0.52	0.46	0.00	0.07	0.46	0.86	2.00

Scatterplot Matrix

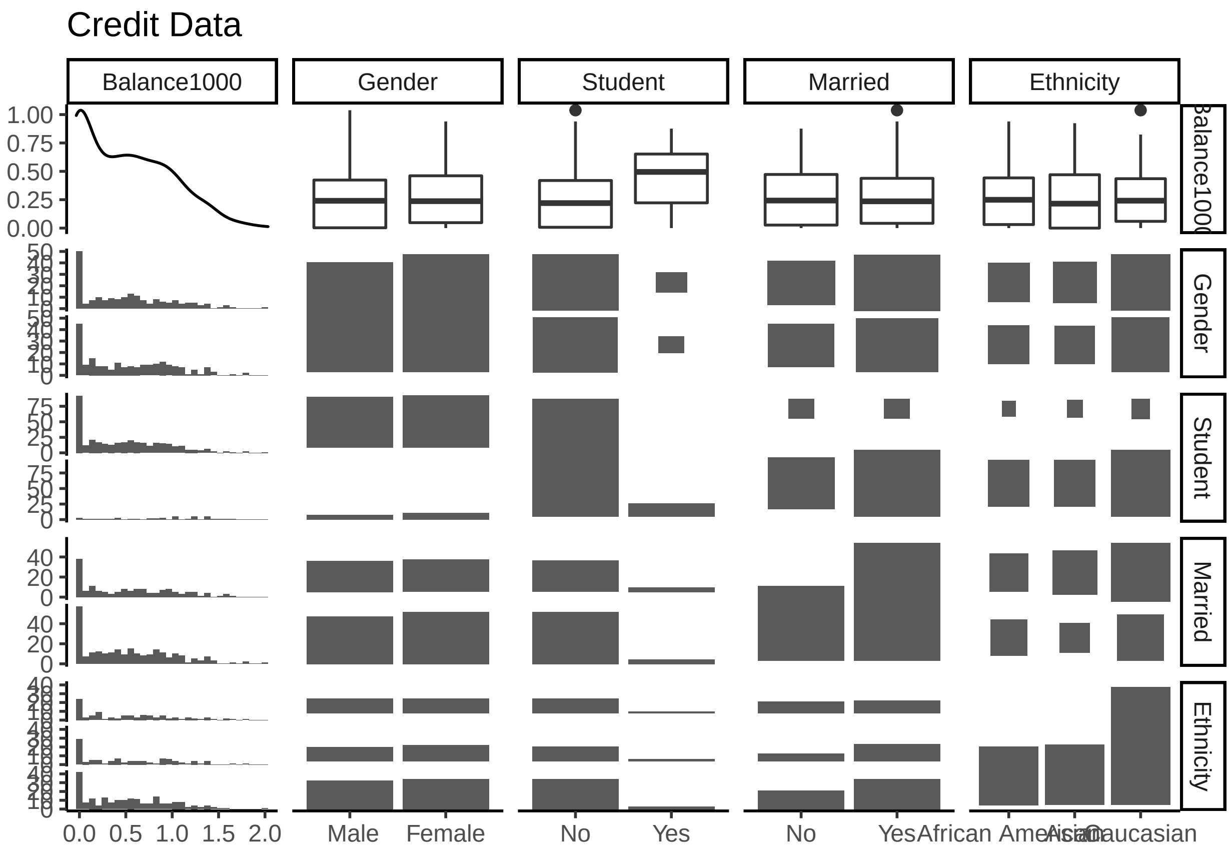
Below is a Scatterplot matrix with student grouping by color. The ggpairs function can do a lot, for now I will leave it at the below.

```r
plot=ggpairs(Credit, columns = c(14,2:6), ggplot2::aes(colour=Student),
             title = "Credit Data by Student Status",
             upper = list(continuous = wrap("cor", size= 2)),
              lower=list(continuous=wrap("points", size=0.1)))
plot
```



Plots with Categorical Variables

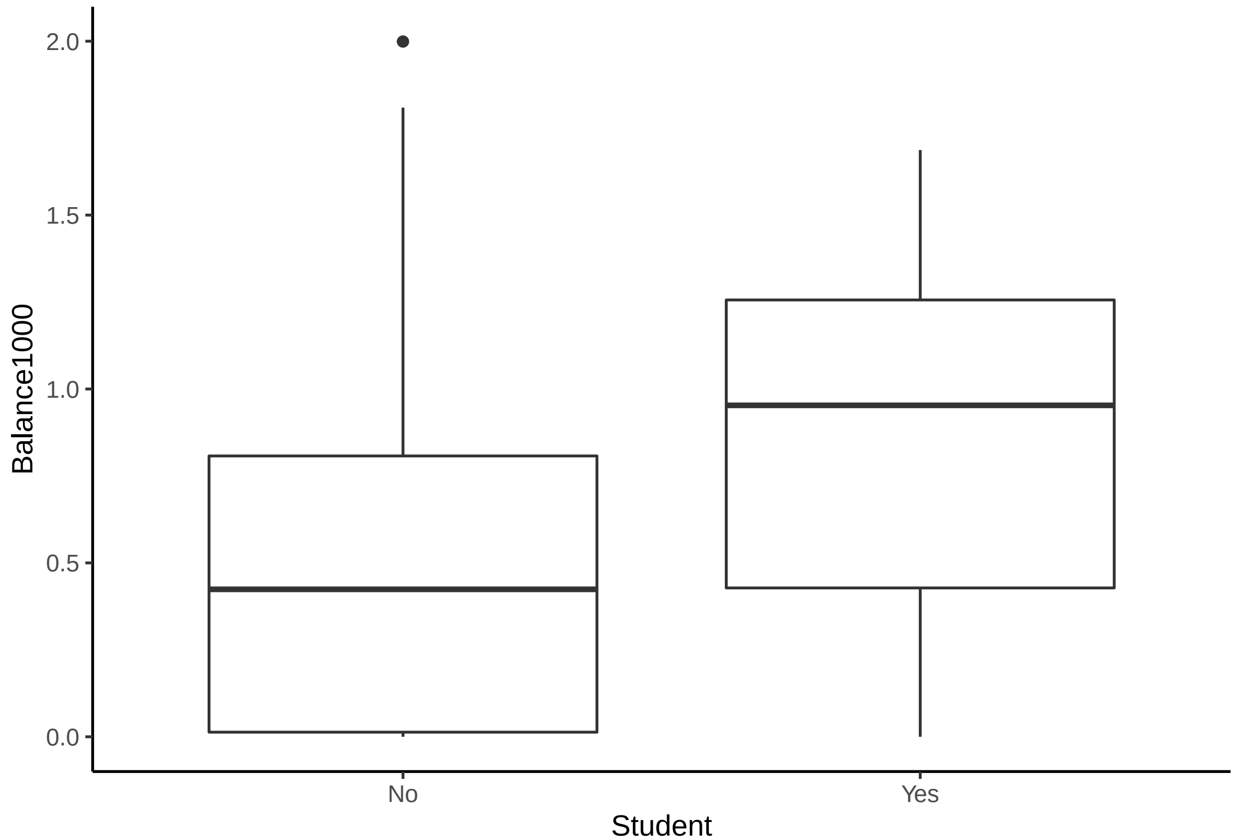
```
plotc=ggpairs(Credit, columns = c(14,8:11),
              title = "Credit Data ",
              lower=list(continuous=wrap("points", size=0.1)))
plotc
```



Student Only Model Using Standard LM and Modern Dive Output

Below illustrates getting fits using standard LM output and Modern Dive Output I will probably mostly use the traditional way. You should be able to read and understand either types of output based on your statistical knowledge.

```r
getPlot(plotc, 1, 3) + guides(fill=FALSE)
```




```r
lm_fit <- lm(Balance ~ Student, data = Credit)
# Get regression table:
get_regression_table(lm_fit, print=TRUE)
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	480.369	23.434	20.499	0	434.300	526.439
StudentYes	396.456	74.104	5.350	0	250.771	542.140

```r
get_regression_summaries(lm_fit, print=T)
```

r_squared	adj_r_squared	mse	rmse	sigma	statistic	p_value	df	nobs
0.067	0.065	196703.8	443.5131	444.626	28.622	0	1	400

```r
summary(lm_fit) #traditional output
```

```
##
## Call:
## lm(formula = Balance ~ Student, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -876.82 -458.82  -40.87  341.88 1518.63
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   480.37      23.43   20.50  < 2e-16 ***
## StudentYes    396.46      74.10    5.35 1.49e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 444.6 on 398 degrees of freedom
## Multiple R-squared:  0.06709,    Adjusted R-squared:  0.06475
## F-statistic: 28.62 on 1 and 398 DF,  p-value: 1.488e-07
```

```
confint(lm_fit)
```

```
##                2.5 %   97.5 %
## (Intercept) 434.2998 526.4390
## StudentYes  250.7707 542.1404
```

Student Only Model Using Stan GLM Defaults

```
lm_stan_default <- stan_glm(Balance ~ Student, data = Credit, refresh=0 )
# Get regression table:
print(lm_stan_default)
```

```
## stan_glm
##  family:       gaussian [identity]
##  formula:      Balance ~ Student
##  observations: 400
##  predictors:   2
## ------
##             Median MAD_SD
## (Intercept) 480.9   22.8
## StudentYes  395.4   69.9
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 445.1   16.1
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

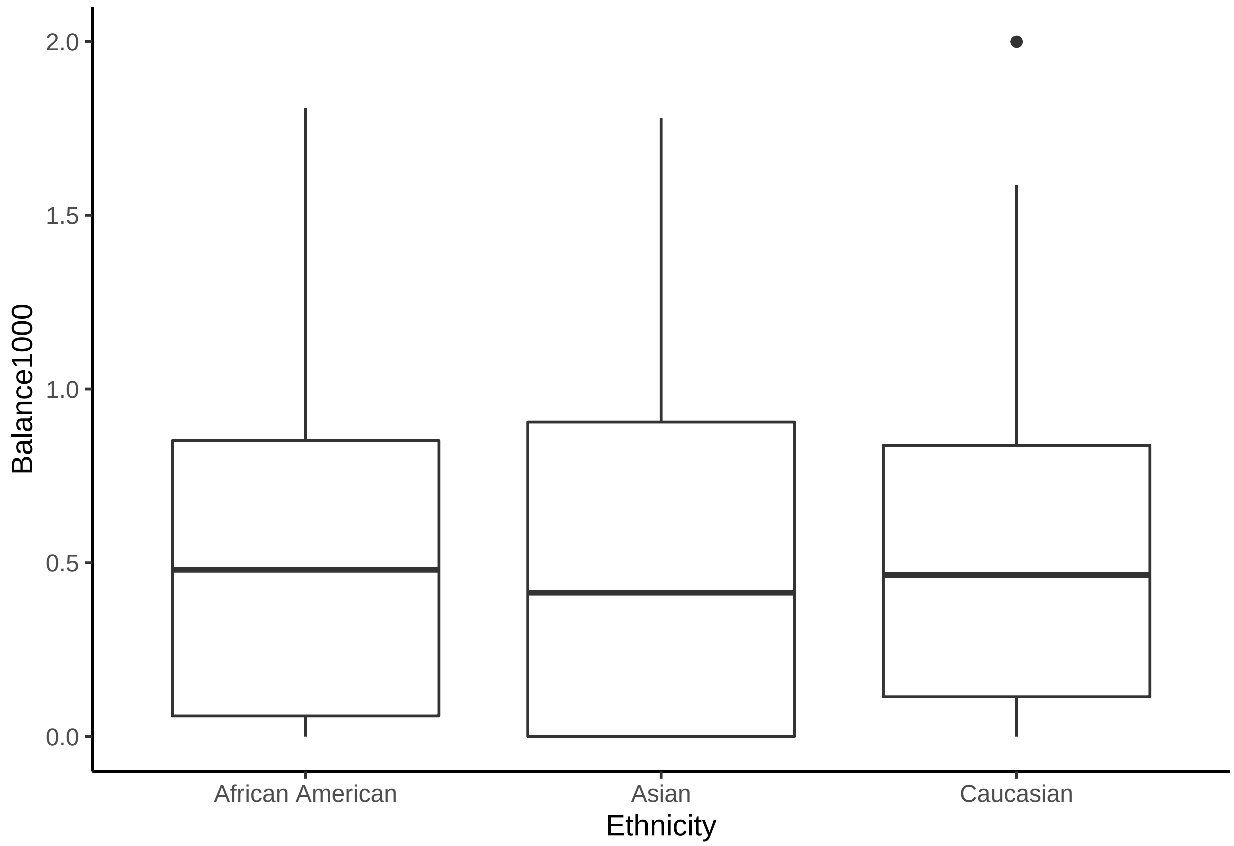
```
sims=as.matrix(lm_stan_default)
quantile(sims[,2],c(0.025,0.975))
```

```
##     2.5%    97.5%
## 254.2726 538.8811
```

We will use lm for now as I want to focus on other learning goals and just do likelihood based analysis.

Qualititative Predictor with More than Two Levels

```
getPlot(plotc, 1, 5) + guides(fill=FALSE)
```




```
lm_fit <- lm(Balance ~ Ethnicity, data = Credit)
# Get regression table:
summary(lm_fit) #traditional output
```

```
##
## Call:
## lm(formula = Balance ~ Ethnicity, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -531.00 -457.08  -63.25  339.25 1480.50
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           531.00      46.32  11.464   <2e-16 ***
## EthnicityAsian        -18.69      65.02  -0.287    0.774
## EthnicityCaucasian    -12.50      56.68  -0.221    0.826
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 460.9 on 397 degrees of freedom
## Multiple R-squared:  0.0002188,  Adjusted R-squared:  -0.004818
## F-statistic: 0.04344 on 2 and 397 DF,  p-value: 0.9575
```

```
confint(lm_fit)
```

```
##                        2.5 %   97.5 %
## (Intercept)         439.9394 622.0606
```

```
## EthnicityAsian     -146.5149 109.1424
## EthnicityCaucasian -123.9350  98.9300
```

Here the baseline category is African American, we can write the fit model (rounding to the nearest dollar) as:

Estimated Balance = \$531 for African Americans = \$ 531-19 = \$ 512 for Asians = \$ 531 - 13 = \$ 518 for Caucasians.

There is not much difference here, based on the small estimated sizes, super small R-squared and large p-value. This does not mean that Ethnicity has no association with credit card debt but there is not evidence of a difference here when ethnicity is looked at individually.

Income and Student

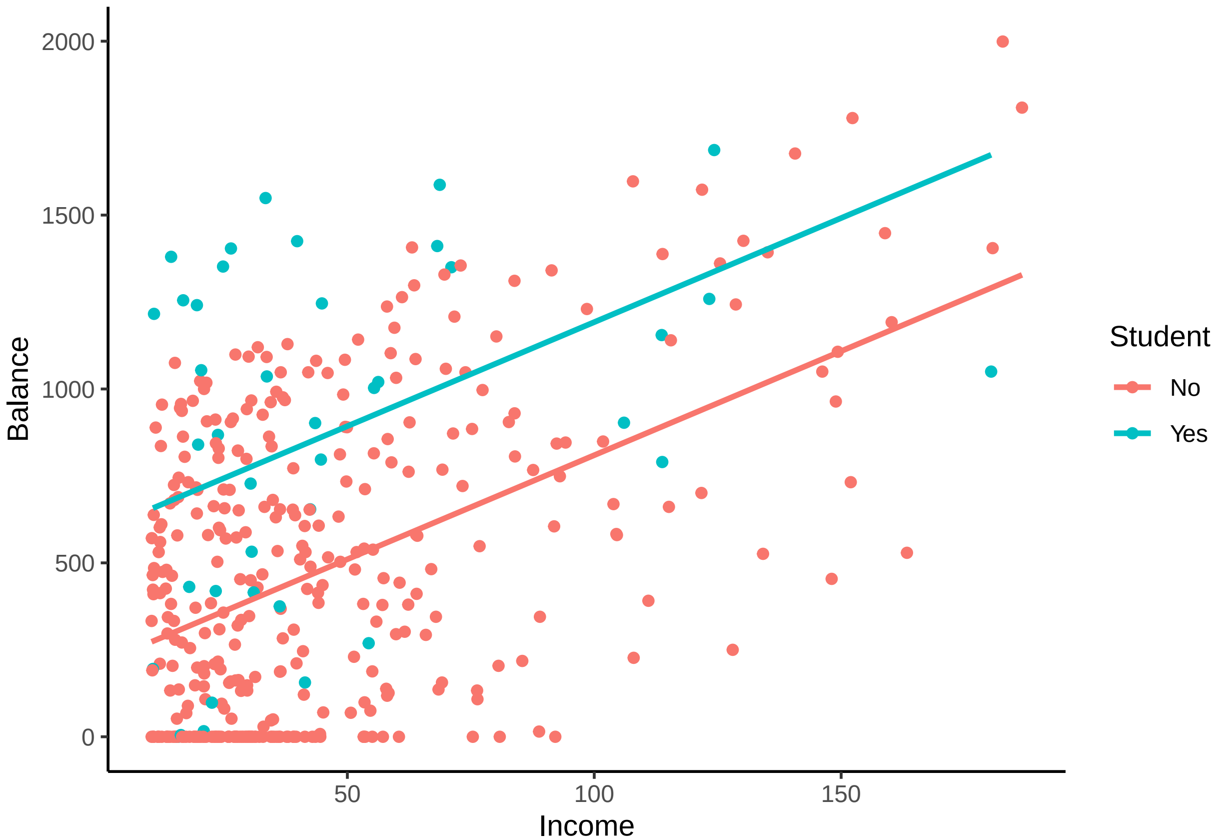
```r
lm_fit <- lm(Balance ~ Income+Student, data = Credit)
# Get regression table:
summary(lm_fit) #traditional output
```

```
##
## Call:
## lm(formula = Balance ~ Income + Student, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -762.37 -331.38  -45.04  323.60  818.28
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 211.1430    32.4572   6.505 2.34e-10 ***
## Income        5.9843     0.5566  10.751  < 2e-16 ***
## StudentYes  382.6705    65.3108   5.859 9.78e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 391.8 on 397 degrees of freedom
## Multiple R-squared:  0.2775, Adjusted R-squared:  0.2738
## F-statistic: 76.22 on 2 and 397 DF,  p-value: < 2.2e-16
```

```r
confint(lm_fit)
```

```
##                  2.5 %     97.5 %
## (Intercept) 147.333469 274.952460
## Income        4.890038   7.078633
## StudentYes  254.272270 511.068807
```

```r
ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  labs(x = "Income", y = "Balance", color = "Student") +
  geom_parallel_slopes(se = FALSE)
```




```
lm_fit <- lm(Balance ~ Income*Student, data = Credit)
# Get regression table:
summary(lm_fit) #traditional output
```

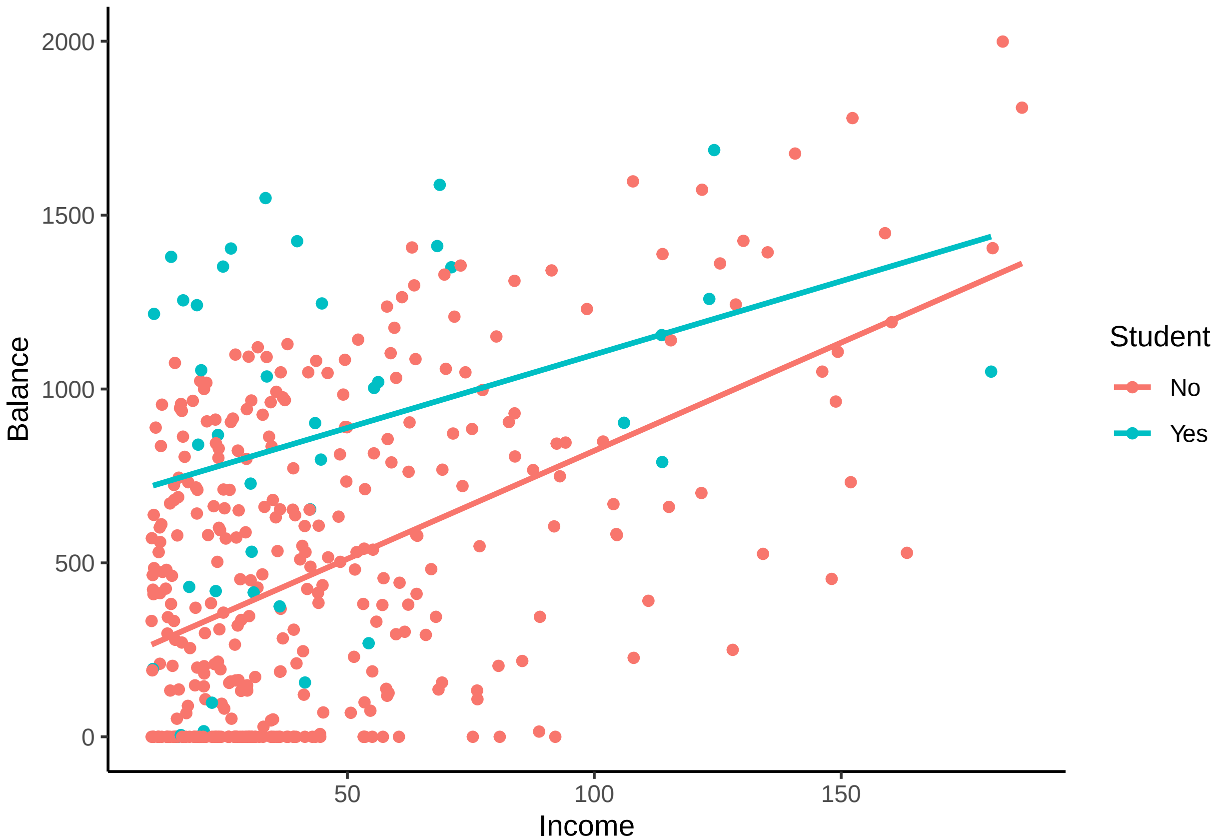
```
##
## Call:
## lm(formula = Balance ~ Income * Student, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -773.39 -325.70  -41.13  321.65  814.04
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         200.6232    33.6984   5.953 5.79e-09 ***
## Income                6.2182     0.5921  10.502  < 2e-16 ***
## StudentYes          476.6758   104.3512   4.568 6.59e-06 ***
## Income:StudentYes    -1.9992     1.7313  -1.155    0.249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 391.6 on 396 degrees of freedom
## Multiple R-squared:  0.2799, Adjusted R-squared:  0.2744
## F-statistic:  51.3 on 3 and 396 DF,  p-value: < 2.2e-16
```

```
confint(lm_fit)
```

```
##                         2.5 %     97.5 %
```

```
## (Intercept)       134.373079 266.873226
## Income              5.054129   7.382208
## StudentYes        271.524196 681.827490
## Income:StudentYes  -5.402743   1.404441
```

```r
ggplot(Credit, aes(x = Income, y = Balance, color = Student)) +
  geom_point() +
  labs(x = "Income", y = "Balance", color = "Student") +
  geom_smooth(method = "lm", se = FALSE)
```



Assumptions of Regression Analysis

Chapter 11 in ROS

1. Validity of the Data- Are the data valid for the question you are trying to answer or the problem you are trying to address? Is the outcome measuring what you are really interested in? Do we have a reasonable set of input variables?
2. Representativeness: Is our sample representative of the populaton of interest?
3. Additivity and Linearity of the response-predictor relationships. (ie the model fits)
4. Independence of error terms.
5. Constant variance of error terms.
6. Residuals are approximately Normal: This is most important if you are doing prediction of future observations and is less important if you are estimating a mean.

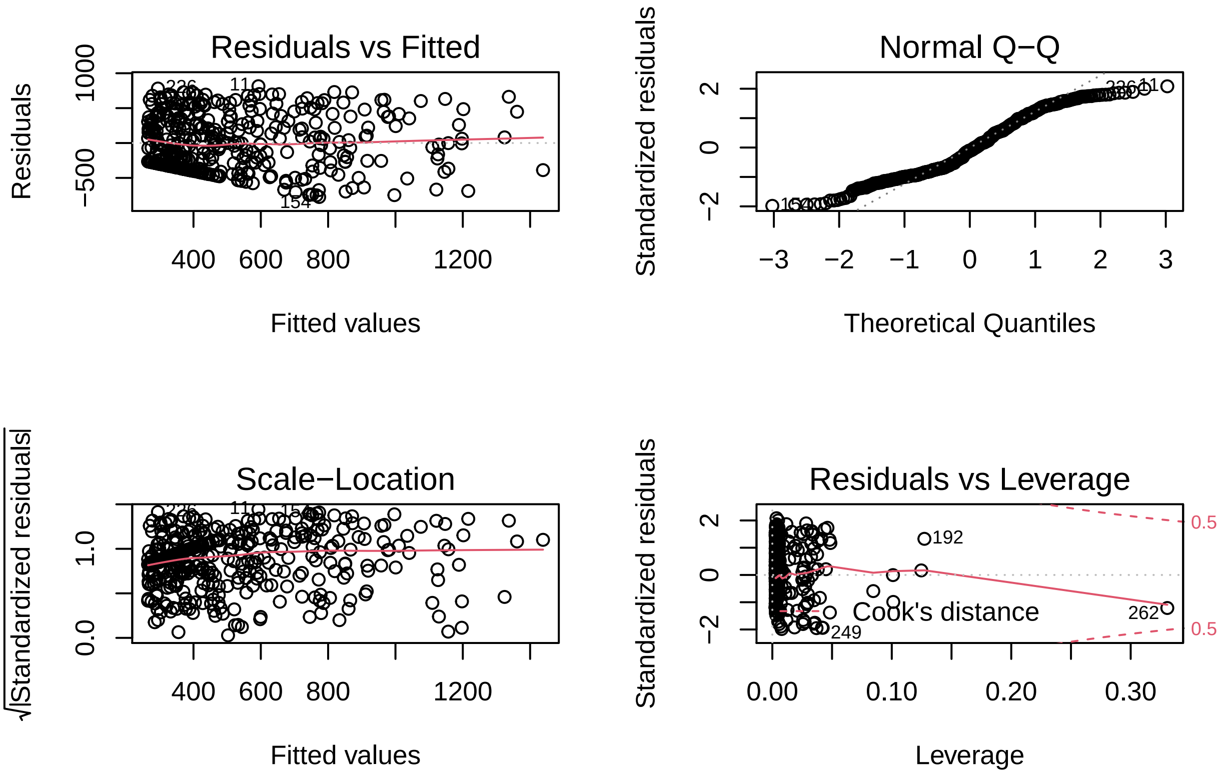
In addition some other things to be concerned about when doing regression are:

1. Outliers.
2. High-leverage points.
3. Multicollinearity.

We can use residual plots to check these conditions and look for systematic lack of fit.

A residual is the actual outcome minus the predicted outcome $e_i = y_i - \hat{y}_i$.

```
par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(lm_fit)
```



The residual vs. fitted plot can help us identify lack of fit. We want this plot to look like a cloud. It does look like there is some interesting things going on. There are probably a lot of people with zero balances and that will impact things. We can check that.

```
mean((Credit$Balance==0))
```

```
## [1] 0.225
```

We see about 23% of people in the data have balances equal to 0.

The Normal QQ plot allows us to check if the residuals are approximately normal. Especially for valid prediction intervals this should also look like a line.

Leverage

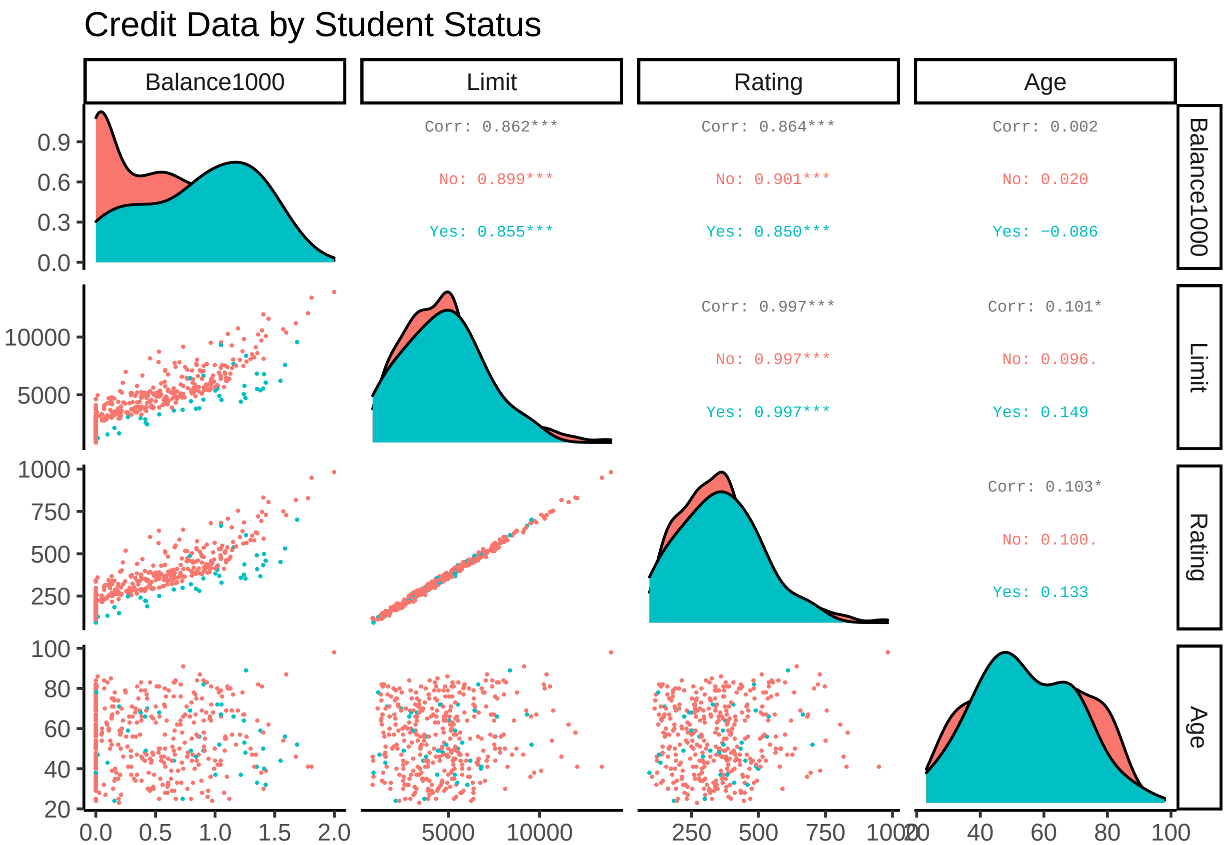
A point is a high leverage point if it is far away from the means of the predictor variables. Note that leverage only depends on the predictors. In the plot above 262 has high leverage and a moderate residual. We see the dashed red lines and those are Cook's D lines. Cook's D is a measure of how much an individual point influences the estimated regression coefficients. If there are points outside the dashed lines we would be concerned because those individual points are really influencing our estimated coefficents.

Non-constant variance

We see some evidence of this in the low end, due to all of those zeros. This reduces the variance down there given you can't get lower than 0. There are models were we use two parts to one part to model the probability of being zero and one to model positive values given non-zero. That might be useful here. We could also expand our model.

Multicollinearity

```
plot=ggpairs(Credit, columns = c(14,3,4,6), ggplot2::aes(colour=Student),
             title = "Credit Data by Student Status",
             upper = list(continuous = wrap("cor", size= 2)),
              lower=list(continuous=wrap("points", size=0.1)))
plot
```



We see that rating and limit have a very high correlation, that means they are highly collinear. Multicollinear means there is a linear combination of the variables that will be near 0 for all the data. (Collinear is just for two variables). Example, of multicollinear variables would be amount of money the change in your pocket is worth, and number of pennies, nickels, dimes and quarters. For most people once you know 4 of those variables you would know the fifth, so these variables are multicollinear. Fit a lm to see how this impacts estimated regression coefficients for the Credit data.

```
lm_fit <- lm(Balance ~ Limit + Rating + Age, data = Credit)
# Get regression table:
summary(lm_fit) #traditional output
```

```
##
```

```
## Call:
## lm(formula = Balance ~ Limit + Rating + Age, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -729.67 -135.82   -8.58  127.29  827.65
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept) -259.51752   55.88219  -4.644 4.66e-06 ***
## Limit          0.01901    0.06296   0.302 0.762830
## Rating         2.31046    0.93953   2.459 0.014352 *
## Age           -2.34575    0.66861  -3.508 0.000503 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 229.1 on 396 degrees of freedom
## Multiple R-squared:  0.7536, Adjusted R-squared:  0.7517
## F-statistic: 403.7 on 3 and 396 DF,  p-value: < 2.2e-16
```

```r
confint(lm_fit)
```

```
##                    2.5 %       97.5 %
## (Intercept) -369.3803781 -149.6546590
## Limit         -0.1047718    0.1427987
## Rating         0.4633782    4.1575409
## Age           -3.6602286   -1.0312747
```

```r
lm_fit <- lm(Balance ~  Rating + Age, data = Credit)
# Get regression table:
summary(lm_fit) #traditional output
```

```
##
## Call:
## lm(formula = Balance ~ Rating + Age, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -733.20 -136.60   -6.52  126.78  836.54
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept) -269.58110   44.80616  -6.017 4.05e-09 ***
## Rating         2.59328    0.07443  34.840  < 2e-16 ***
## Age           -2.35078    0.66764  -3.521  0.00048 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 228.8 on 397 degrees of freedom
## Multiple R-squared:  0.7535, Adjusted R-squared:  0.7523
## F-statistic: 606.9 on 2 and 397 DF,  p-value: < 2.2e-16
```

```r
confint(lm_fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) -357.668110 -181.494091
```

```
## Rating         2.446945    2.739611
## Age           -3.663334   -1.038225
```

```r
lm_fit <- lm(Balance ~  Limit +Rating, data = Credit)
# Get regression table:
summary(lm_fit) #traditional output
```

```
##
## Call:
## lm(formula = Balance ~ Limit + Rating, data = Credit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -707.8 -135.9   -9.5  124.0  817.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -377.53680   45.25418  -8.343 1.21e-15 ***
## Limit          0.02451    0.06383   0.384   0.7012
## Rating         2.20167    0.95229   2.312   0.0213 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 232.3 on 397 degrees of freedom
## Multiple R-squared:  0.7459, Adjusted R-squared:  0.7447
## F-statistic: 582.8 on 2 and 397 DF,  p-value: < 2.2e-16
```

```r
confint(lm_fit)
```

```
##                    2.5 %       97.5 %
## (Intercept) -466.5045792 -288.5690115
## Limit         -0.1009816    0.1500104
## Rating         0.3295030    4.0738414
```

```r
lm_fit <- lm(Balance ~  Rating, data = Credit)
# Get regression table:
summary(lm_fit) #traditional output
```

```
##
## Call:
## lm(formula = Balance ~ Rating, data = Credit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -712.28 -135.32   -9.58  125.67  829.04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -390.84634   29.06851  -13.45   <2e-16 ***
## Rating         2.56624    0.07509   34.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 232.1 on 398 degrees of freedom
## Multiple R-squared:  0.7458, Adjusted R-squared:  0.7452
## F-statistic:  1168 on 1 and 398 DF,  p-value: < 2.2e-16
```

```r
confint(lm_fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) -447.993365 -333.699319
## Rating         2.418619    2.713861
```

Some ways we can deal with multicollinearity include not using all the variables, feature construction such as averaging the variables, and shrinkage methods using priors or techniques like ridge regression which shrink the coeefficients and reduce their standard errors.