PM592: Regression Analysis for Health Data Science
Lab 4 – Regression Diagnostics
Data Needed: *Class Survey Data*

Outline

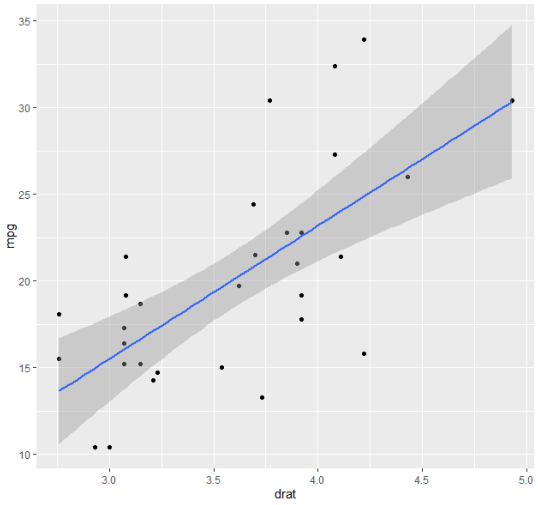
- Model Diagnostics
- Variable Manipulation: Stringr
- Factors

1. Model Diagnostics

1.1. Plot()

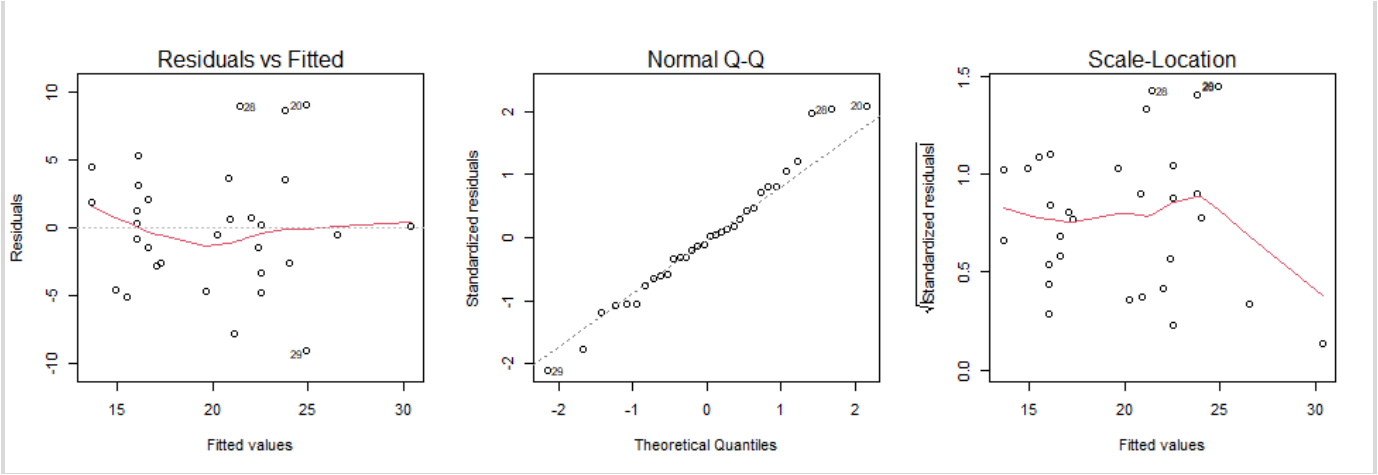
1.1.1. When the plot() command sees a lm object, it will draw plots related to residual diagnostics.

1.1.2. Let's use the "mtcars" data set and examine the relationship between MPG and drat (rear axle ratio).



1.1.3. The par() function sets graphical parameters that are used in plotting. You can specify mfrow = c(nrow, ncol) to combine figures into a nrow x ncol matrix, filling in by row. Similarly, specifying mfcol = c(nrow, ncol) will do the same, but filling in by column.

```
> par(mfrow = c(1, 3))
> lm(mpg ~ drat, data = mtcars) %>%
+   plot()
```



1.1.4. Here we see some slight deviations from the assumptions. Note when the sample size is smaller, any slight departure from the assumptions can be seen clearly. For example, homoscedasticity is violated for high values, but this is because the fit point is directly on the regression line. Here, we can use our judgment to determine that the assumptions are generally met.

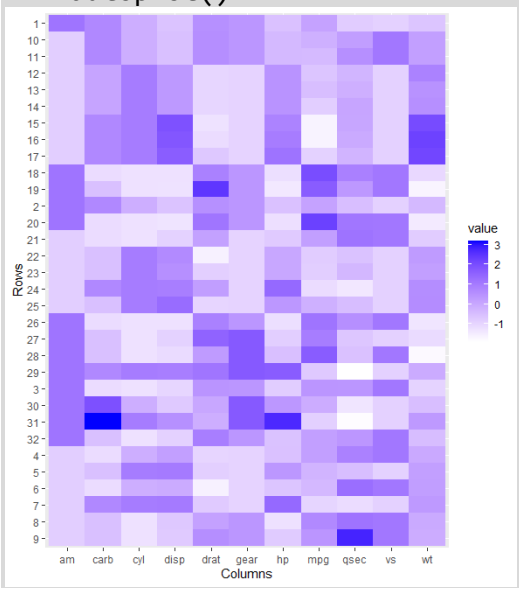
1.2. ggfortify

1.2.1. The ggfortify package will allow ggplot2 to draw regression diagnostics. There's nothing substantially different about this other than the drawings will look nicer and can be manipulated using ggplot's rules and syntax.

1.2.2. In addition to linear models, ggfortify will handle objects from different analyses such as time series and survival models.

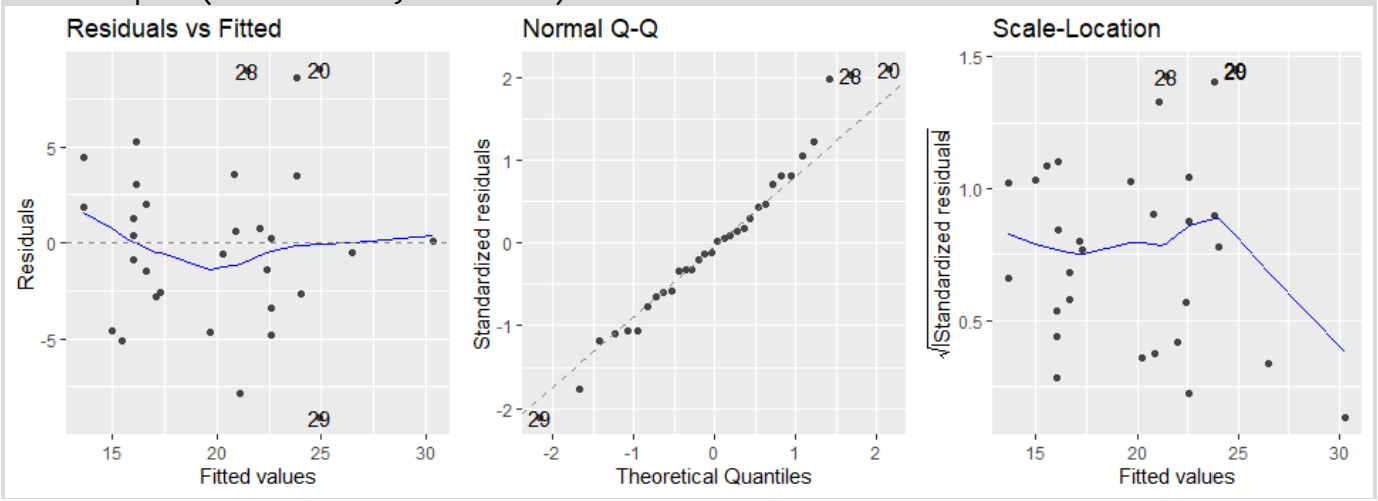
1.2.3. In the following code, we produce a heat map of values (this is not a regression diagnostic but can be helpful in certain situations).

```
> mtcars %>%
+   select_if(is.numeric) %>%
+   scale() %>%
+   autoplot()
```



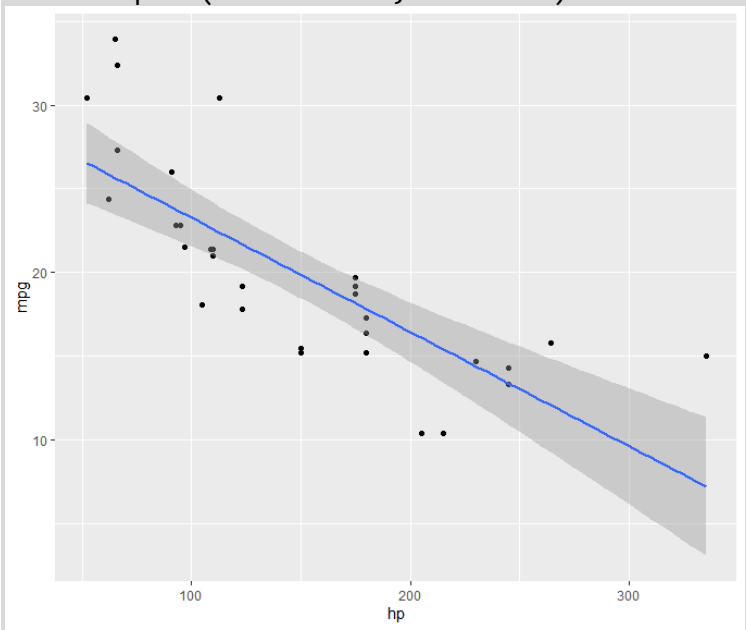
1.2.4. The plot() function for a lm object actually returns 6 plots. So far we have focused on the first 3 and will continue to do so. Here, we specify that ggfortify return the first 3 plots.

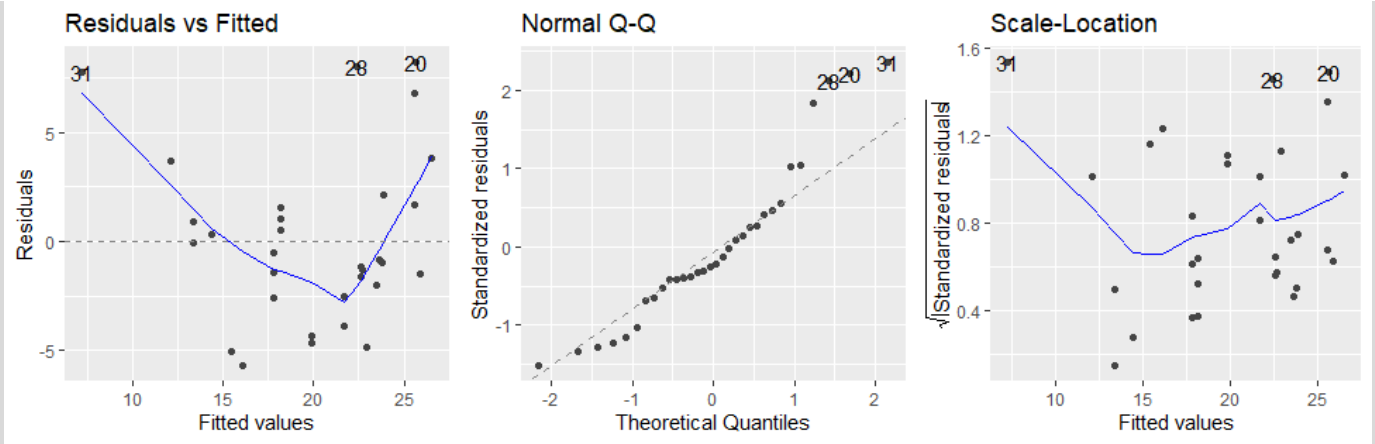
```
> lm(mpg ~ drat, data = mtcars) %>%
+   autoplot(which = 1:3, ncol = 3)
```



1.2.5. Now let's look at the relationship between MPG and HP (horse power):

```
> mtcars %>%
+   ggplot(aes(x = hp, y = mpg)) +
+   geom_point() +
+   geom_smooth(method = "lm")
`geom_smooth()` using formula 'y ~ x'
> lm(mpg ~ hp, data = mtcars) %>%
+   autoplot(which = 1:3, ncol = 3)
```

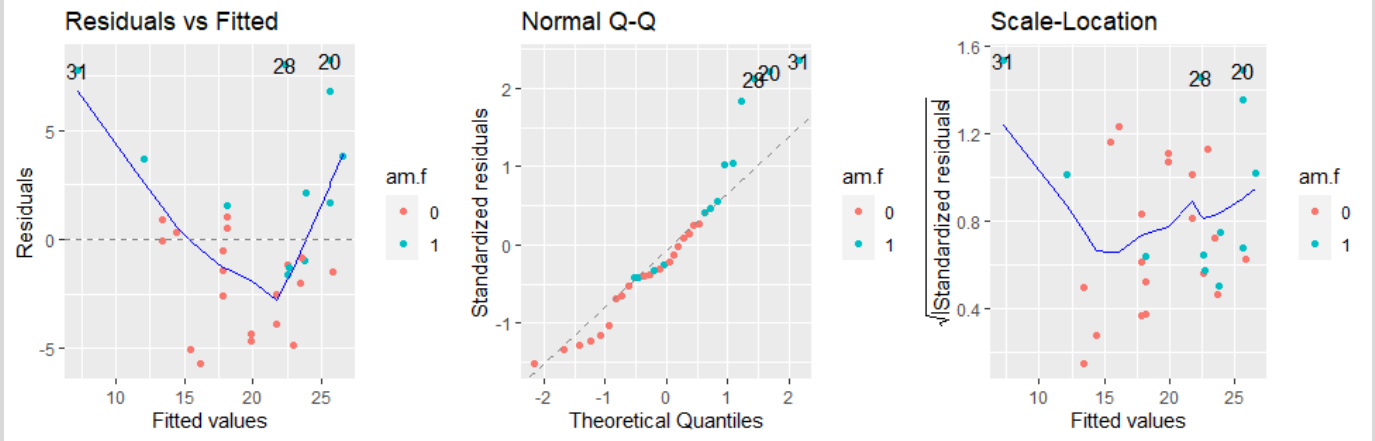




1.2.6. We see the fit is worse; there are violations of linearity and normality, in particular.

1.2.7. We can use ggfortify to color the residuals by variables in our data set. Perhaps this can help explain why we are seeing violations of our assumptions. In the following code, we color the points by "am.f", which is an indicator of whether the car has automatic transmission (1=yes, 0=no).

```
> lm(mpg ~ hp, data = mtcars) %>%
+   autoplot(which = 1:3,
+            ncol = 3,
+            data = mtcars,
+            colour = 'am.f')
```



1.2.8. Examine the plot on the left. The residuals are mostly positive for automatic transmissions (blue) and mostly negative for manual transmissions (pink). This is telling us that our model seems to be underestimating the MPG of automatic transmissions and overestimating the MPG of manual transmissions. Clearly, incorporating this variable into our analysis will help our model. We will discuss this further next week.

2. Variable Manipulation

2.1. The "mtcars" data set uses R's row names feature. When a data frame is converted to a tibble, the row names are dropped. We can prevent this from happening by specifically converting the row names to an individual column. We can then rename the new "rowname" variable to something more descriptive.

```
> head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1

> mtcars <-
+   mtcars %>%
+   rownames_to_column()

> head(mtcars)
            rowname  mpg cyl disp  hp drat    wt  qsec vs am gear carb
1         Mazda RX4 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
2     Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
3        Datsun 710 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
4    Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
5 Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
6           Valiant 18.1   6  225 105 2.76 3.460 20.22  1  0    3    1

> mtcars <-
+   mtcars %>%
+   rownames_to_column() %>%
+   rename(carname = rowname)

> head(mtcars)
            carname  mpg cyl disp  hp drat    wt  qsec vs am gear carb
1         Mazda RX4 21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
2     Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
3        Datsun 710 22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
4    Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
5 Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
6           Valiant 18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

2.2. We can gain additional information from the new "carname" variable. For instance, each car may not be independent of the others; this variable contains information on both the *make* and the *model* of the car.

2.3. If we want to extract the first word from this string and place it in a new "make" variable, there are a couple ways we can do this:

2.3.1. We can use regex commands and the grep() family of variables. While powerful, this approach has a steeper learning curve

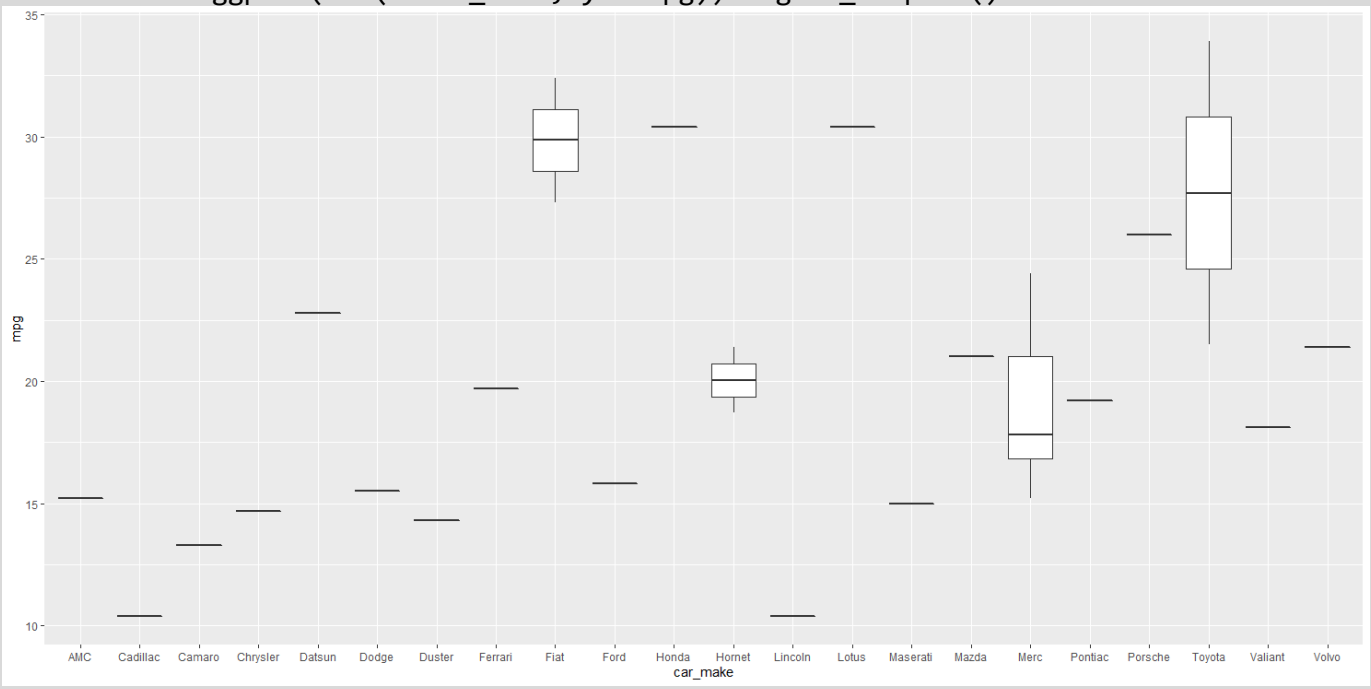
2.3.2. We can use the stringr package, which is a part of the Tidyverse. (https://stringr.tidyverse.org/).

```
> mtcars <-
+   mtcars %>%
+   mutate(
+     car_make = word(carname, 1),
+     car_model = word(carname, 2, -1)
+     )
```

```
> mtcars %>% select(carname, car_make, car_model)
# A tibble: 32 x 3
   carname           car_make car_model
   <chr>             <chr>    <chr>
 1 Mazda RX4         Mazda    RX4
 2 Mazda RX4 Wag     Mazda    RX4 Wag
 3 Datsun 710        Datsun   710
 4 Hornet 4 Drive    Hornet   4 Drive
 5 Hornet Sportabout Hornet   Sportabout
 6 Valiant           Valiant  NA
 7 Duster 360        Duster   360
 8 Merc 240D         Merc     240D
 9 Merc 230          Merc     230
10 Merc 280          Merc     280
# ... with 22 more rows
```

2.4. This isn't much use to us as several car makes have only one observation in this data set. However, we do see a couple things, such as: 1) Fiat generally has really good MPG, 2) Mercedes and Hornet generally have poorer MPG, and 3) Toyota has a car with the best MPG but also some cars with mediocre MPG.

```
> mtcars %>% ggplot(aes(x=car_make, y = mpg)) + geom_boxplot()
```



3. **Factors**

3.1. When creating a factor variable, the factor levels will be ordered either numerically or alphabetically.

3.2. We created "cyl.f", a factor variable for the number of cylinders. The responses are 4, 6, and 8; we see that the factor variable is ordered numerically.

```
> str(mtcars$cyl.f)
 Factor w/ 3 levels "4","6","8": 2 2 1 2 3 2 3 1 1 2 ...
```

3.3. If we perform a linear regression with dummy variables for these groups, we see that the estimated MPG for 4-cylinder cars is 26.66. The mean MPG for 6-cylinder cars is 6.92 *lower* than for 4-cylinder cars (p<.001); and the mean MPG for 8-cylinder cars is 11.56 *lower* than for 4-cylinder cars (p<.001).

```
> lm(mpg ~ cyl.f, data = mtcars) %>%
+   summary()

Call:
lm(formula = mpg ~ cyl.f, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2636 -1.8357  0.0286  1.3893  7.2364

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  26.6636     0.9718  27.437  < 2e-16 ***
cyl.f6       -6.9208     1.5583  -4.441 0.000119 ***
cyl.f8      -11.5636     1.2986  -8.905 8.57e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.223 on 29 degrees of freedom
Multiple R-squared:  0.7325,  Adjusted R-squared:  0.714
F-statistic:  39.7 on 2 and 29 DF,  p-value: 4.979e-09
```

3.4. Suppose we want 8-cylinder to be the reference group. We can accomplish this recoding two ways.

3.4.1. First, we can manually recode all values. This is done by specifying “levels =” when creating the factor.

3.4.2. Second, we can simply change the reference category using the relevel() function with the “ref =” option.

3.4.3. There’s a subtle difference between these two methods, as you can see below.

```
> mtcars <-
+   mtcars %>%
+   mutate(
+     cyl.rf1 = factor(cyl, levels = c(8, 6, 4)),
+     cyl.rf2 = relevel(cyl.f, ref = "8")
+     )

> mtcars %>%
+   select(starts_with("cyl.rf")) %>%
+   str()
tibble [32 x 2] (S3: tbl_df/tbl/data.frame)
 $ cyl.rf1: Factor w/ 3 levels "8","6","4": 2 2 3 2 1 2 1 3 3 2 ...
 $ cyl.rf2: Factor w/ 3 levels "8","4","6": 3 3 2 3 1 3 1 2 2 3 ...
```

3.4.4. However, our regression results will not be affected by this.

```
> lm(mpg ~ cyl.rf1, data = mtcars) %>%
```

```
+   summary()

Call:
lm(formula = mpg ~ cyl.rf1, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2636 -1.8357  0.0286  1.3893  7.2364

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  15.1000     0.8614  17.529  < 2e-16 ***
cyl.rf16      4.6429     1.4920   3.112  0.00415 **
cyl.rf14     11.5636     1.2986   8.905 8.57e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.223 on 29 degrees of freedom
Multiple R-squared:  0.7325,  Adjusted R-squared:  0.714
F-statistic:  39.7 on 2 and 29 DF,  p-value: 4.979e-09

> lm(mpg ~ cyl.rf2, data = mtcars) %>%
+   summary()

Call:
lm(formula = mpg ~ cyl.rf2, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-5.2636 -1.8357  0.0286  1.3893  7.2364

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  15.1000     0.8614  17.529  < 2e-16 ***
cyl.rf24     11.5636     1.2986   8.905 8.57e-10 ***
cyl.rf26      4.6429     1.4920   3.112  0.00415 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.223 on 29 degrees of freedom
Multiple R-squared:  0.7325,  Adjusted R-squared:  0.714
F-statistic:  39.7 on 2 and 29 DF,  p-value: 4.979e-09
```

Lab 4 Exercises

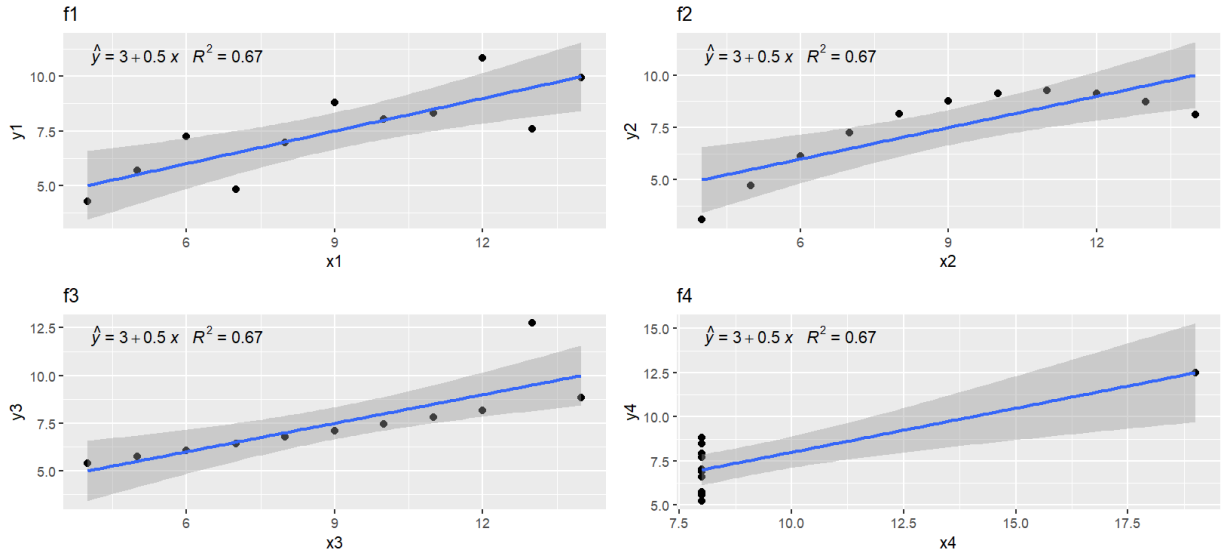
Objective(s):	Assess regression diagnostics, practice setting and merging data sets
Datasets Required:	`lab1q1.dat, lab1q2.dat, lab2q1.dat, lab2q2.dat`

Part I – We will use the Anscombe data set. Load the data by using data(anscombe). You will also need the cowplot and ggpmisc packages installed.

1) Create 4 scatter plots with a linear-fit line. Title your scatter plots (f1, f2, f3, f4). F1 should have x1 on the X-axis and y1 on the Y-axis, f2 should have the relationship between y2 and x2, and so forth. Also: Add the regression equation and R^2 value to each graph. Add the following in ggplot:

```
stat_poly_eq(formula = y ~ x,
             eq.with.lhs = "italic(hat(y))~`=`~",
             aes(label = paste(..eq.label.., ..rr.label.., sep = "~~~")),
             parse = TRUE)
```

   a) After obtaining f1 through f4, combine them into one ggplot object using plot_grid(f1, f2, f3, f4) and paste your graphs here.



   b) In which graph is the variance of the Y variable explained the best by the X variable?

      The proportion of the variance in Y that is explained by X (given by R^2) is the same in all four graphs

   c) In which graph is the relationship between X and Y the strongest?

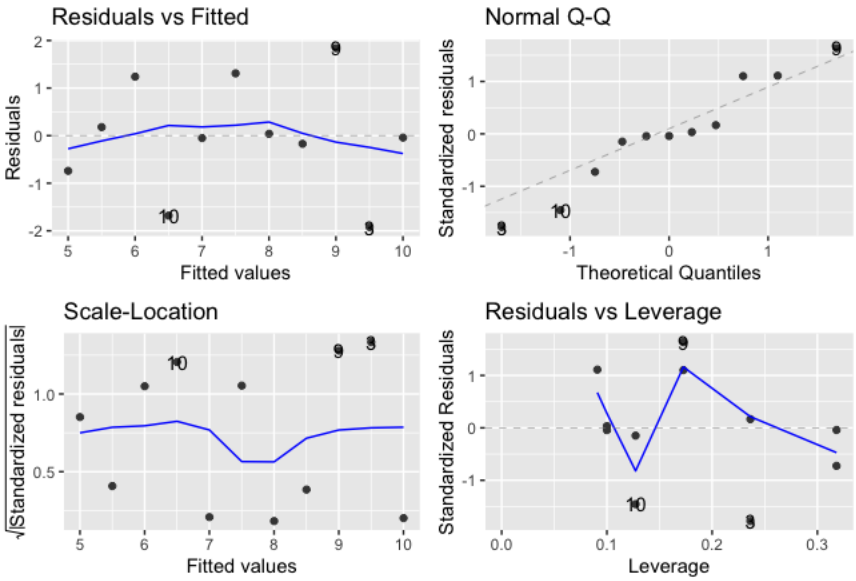
      The relationship between X and Y, given by the slope of the regression line is the same in all four graphs

   d) Run 4 linear regressions and report in which one the p-value is the lowest.

      The p-value was about the same for all four models, at about 0.0021

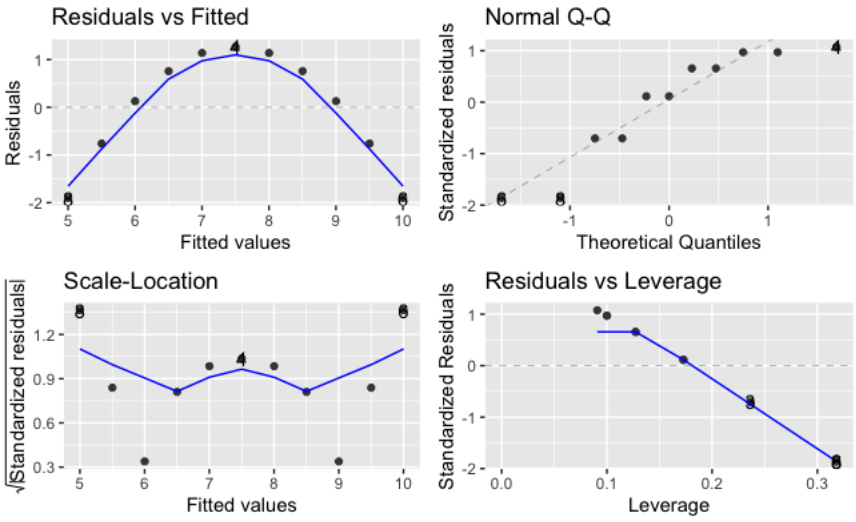
2) For each set of relationships (1 through 4), evaluate the assumptions of linear regression (assume independence). For each of the 4 relationships, provide a set of plots from ggfortify to support your responses

f1:



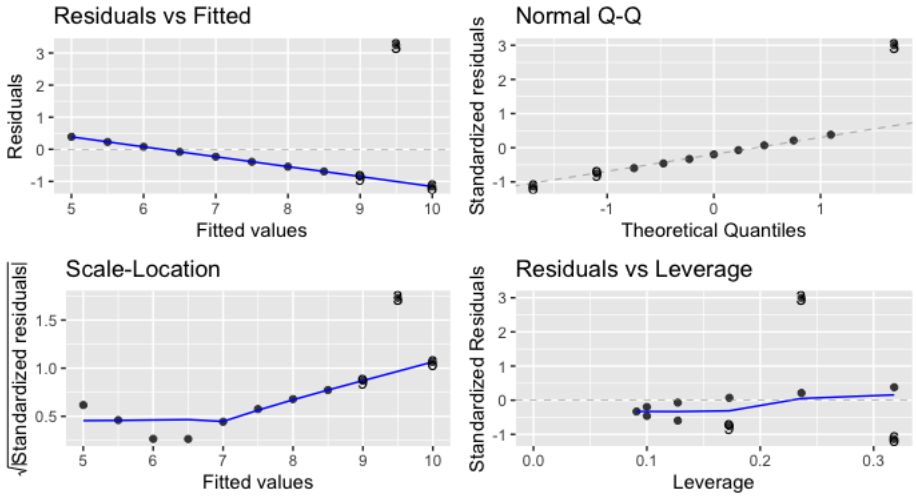
The assumptions of linearity appear to be held.

f2:



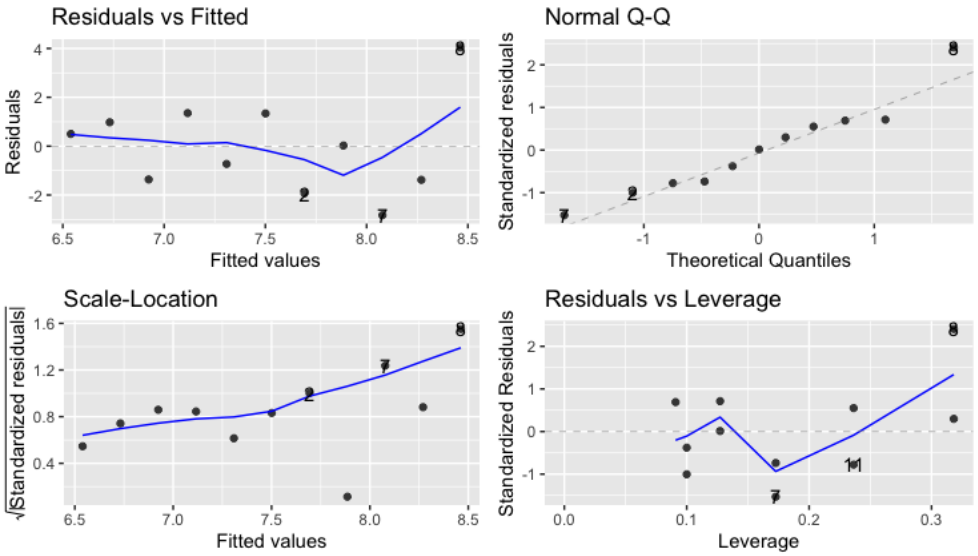
In f2, the assumption of linearity is violated.

f3:



Outlier can be seen in the top left and bottom left plots. The residual vs fitted plot looks like it is decreasing, hinting at deviation from normality.

f4:



The assumptions of normality look more or less met in these diagnostic plots.

3) For which set of relationships do you most trust the regression line to explain the association between X and Y? For each of the other methods, explain why you don't trust the regression line to explain the association.

   The regression line in f1 seems to be the most trustworthy.

   In f3 and f4, the presence of outliers in X and Y respectively, makes the regression line to be less trustworthy.

   In f2, the assumption of normality is violated, making the regression not trustworthy

Part II – Students in a previous class filled out 2 questionnaires, which are stored in 4 separate ASCII files. The name of the files and their corresponding contents are summarized in Table 1. The data files are summarized in Tables 2 and 3.

Table 1: Name of the file and contents.

Name of the file	Contents
lab1q1.dat	Questionnaire #1 from Early Lab
lab2q1.dat	Questionnaire #1 from Late Lab
lab1q2.dat	Questionnaire #2 from Early Lab
lab2q2.dat	Questionnaire #2 from Late Lab

Table 2: Data Description for lab1q1.dat and lab2q1.dat

Variable	Beginning column	Description/Value
ID	1	Four-digit ID
Sex	6	M or F
HtFeet	8	Height in feet (i.e. for 5'7" "5" is the value)
HtInches	9	Height in inches (i.e. for 5'7" "7" is the value)
Race	12	Race/Ethnicity 1 = Asian; 2 = Hispanic; 3 = White; 4 = Other
BirthM	14	Birth Month
BirthD	16	Birth Day
LabNum	18	1 = Early Lab (11:00-1:00); 2 = Late Lab (3:00-5:00)

Table 3: Data Description for lab1q2.dat and lab2q2.dat

Variable	Beginning column	Description/Value
ID	1	Four-digit ID
Transportation	6	How to you come to school? 1 = Car; 2 = Walk; 3 = Public Transportation 4 = Bike
GasPrice	8	How much per gallon of gas did you pay when you last filled up your car?
SUV	13	Type of your car. 1 = SUV; 0 = Non-SUV
CarMake	15	What make is your car?
TankVol	26	How many gallons of gas can your gas tank hold?
TankLast	29	How many miles can you drive on a whole tank of gas?
City	33	Which city do you live?
OneWay	50	How many miles from school to home (1 way?)
SchTimes	56	How many times do you drive to school each week?
LabNum	58	1 = Early Lab (11:00-1:00); 2 = Late Lab (3:00-5:00)

4) Combine the four ASCII files into one single R Tibble. You'll have to use the "read_fwf" command, which is appropriate for fixed-with data files. You'll have to specify the widths of each variable and the variable names. See the following code for an example of how to read-in lab1q1.dat:

```
read_fwf("lab1q1.dat",
         fwf_widths(
           c(5, 2, 1, 3, 2, 2, 2, 2),
           c("id", "sex", "htfeet", "htinches",
             "race", "birthm", "birthd", "labnum")),
         na = "."
)
```

   a) Report the dimensions of your resulting data set.

      36 rows by 17 columns (with the duplicate column removed)

5) Use regression analysis to fit the model:

$$\hat{Y}_{TANKVOL} = \beta_0 + \beta_1 X_{TANKLAST}$$

   a) What is the resulting model?

$$\hat{Y}_{TANKVOL} = 12.15 + 0.01 X_{TANKLAST}$$

   b) What is the mean tank volume for a car that can drive for 380 miles on a full tank of gas?

      15.95 gallons

   c) Is the number of miles a car can be driven on a whole tank of gas a significant predictor of tank size? Provide statistical evidence (test statistic and p-value) to support your decision.

      With a t-statistic of 1.097 and a p-value of 0.287123, it appears that the number of miles a car can be driven on a whole tank of gas is not a significant predictor of tank size. There is not enough evidence to reject the null hypothesis that the number of miles a car can be driven on a whole tank of gas is not a significant predictor of tank size.

6) Create a factor variable for transportation type and provide descriptive labels for the factor. Then, determine if the one-way commuting distance is related to transportation type.

   a) What is the overall p-value for whether one-way commuting distance is related to transportation type?

      p-value: 0.09347 (F-test)

   b) How much of the variance in one-way commuting distance is explained by transportation type?

      Multiple R-squared: 0.2148 (approximately 21% of the variance in one-way commuting distance is explained by transportation type)

   c) On average, what is the difference in one-way commuting distance for those who drive cars vs. those who take public transportation?

      -4.729 (coefficient for public transportation)

   d) On average, what is the difference in one-way commuting distance for those who walk vs. those who take public transportation? Compute this by changing the factor reference group to those

who walk and re-running the regression.

10.7833 (coefficient for public transportation after `relevel()` applied to transportation column in data set)