Voluntary assignment - example solution with some comments

Jaromír Mazák & Aleš Vomáčka

1. 7. 2021

# How to work with this document

This is only an example solution. It does not mean that your different solution cannot be perfectly adequate, too. The text on grey background is the R code we used to produce our solution. It also includes additional comments which may be of interest.

This is a *long version* which far exceeds what we expected you to submit, but it may be a welcome learning material. The added value of this longer version is that we show more of our thinking behind preparing the material. However, it may feel a bit cumbersome. Worry not, we also produced a *short version* which indicates the expected extent of your submission and is much more straightforward.

(Note: Detailed assignment instructions were provided in a separate document and are not repeated here.)

# STEP 1: Preparation

## Load libraries

# Always keep all libraries used in the script at the top of the script.   
  
library(tidyverse) # for core tidyverse packages including dplyr and ggplot2  
library(here) # for smart path referencing  
library(patchwork) # for arranging graphs next to each other  
library(ggeffects) # for ggpredict, i.e. for plotting marginal effects  
  
options(scipen = 999) # turn off scientific notaiton (produces easier to read tables)

## Make sure you know and keep in mind what the task is

Make sure you know what exactly is your task. In this case, it is answering the following question: Is the accuracy of polls (variable *error*) different depending on the methodology of data collection (variable *methodology*)? However, there have been specific requirements in the instructions as of what your solution should include. Keep that all in mind.

## Data import

polls = read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/pollster-ratings/raw-polls.csv")  
  
polls %>% count(methodology) %>% arrange(desc(n)) # we only analyse "IVR", "Live Phone", "Online" becuase (a) they are most frequent, (b) the combination catgories would be harder to interpret.

## methodology n  
## 1 Live Phone 5603  
## 2 IVR 2518  
## 3 Online 1586  
## 4 IVR/Online 529  
## 5 IVR/Live Phone 152  
## 6 Online/Text 86  
## 7 Live Phone/Online 84  
## 8 IVR/Online/Text 82  
## 9 IVR/Text 59  
## 10 Text 30  
## 11 Mail 22  
## 12 IVR/Live Phone/Text 7  
## 13 Live Phone/Text 6  
## 14 IVR/Live Phone/Online 4  
## 15 Live Phone/Online/Text 4  
## 16 Face-to-face 3  
## 17 IVR/Live Phone/Online/Text 1

polls = subset(polls, methodology == c("IVR", "Live Phone", "Online"))  
  
polls = as\_tibble(polls)  
  
# You can look at the variables description here - need to uncomment below:  
# browseURL("https://github.com/fivethirtyeight/data/tree/master/pollster-ratings")

# STEP 2: Variable selection

### Dependent variable:

* *error* (“Absolute value of the difference between the actual and polled result.”)

### Main independent variable:

* *methodology* (“Methodology used to conduct this poll. One or more of the following values: … only relevant values for us are *IVR*, i.e. Interactive voice response, otherwise known as automated polls or”robopolls"; *Live Phone*, i.e., Live telephone interviews, may or may not include calls to cell phones ; and *Online*, i.e., Poll conducted by Internet; generally this mean by web browser, or application-based polling of mobile phones … see the filtering in data import section)

### Other independent variables:

This is part of your solution. Generally, we include other independent variables to control for them, i.e., we include those variables which we expect to have confounding influence on the relationship we are interested in. Below, you find our solution and arguments. Obviously, we first had to run a lot of simple descriptive analyses to get an ideas what each variable is about. We do not include those here, but that is something you should also do.

*In our view, you most likely should include the following:*

* *year* - polling standards may have changed since 1998, so it makes good sense to include it
* *type\_simple* - it differentiates the following US elections: president primary, president general, senators general, house of representatives general, and governors general election. We think it makes good sense to expect that different election types are easier or more difficult to predict. So we should control for it.
* *polldate* and *electiondate* - by themselves, these variables are not very interesting. In additions, we already include the election year. But it could be very important to consider the difference between the two, i.e. the gap between the polling and the actual election as polling further away from the elections is expected to be less precise in terms of predicting the actual result.
* *samplesize* - Sample size is important for each individual poll’s estimate (specifically, its precision), so we need to include it in the model. Otherwise, if any of the polling methods is associated with different sample sizes than the other two, the results could just be reflecting the different precision of different sample sizes, not the different perfomance of each method.

*We also think it could be good to include the following:*

* *partisan* - on theoretical basis, it could make sense to include whether the poll is independent or commissioned by a specific party. However, the proportion of “partisan” polls is so small, that it is unlikely to influence the results very much. Also, the source webpage warns that the coding may be inconsistent here. Still, we include this variable as a control because it could help make a slightly better model and it should not cause any harm.

*Debatable are:*

* *race* (or race\_id) - it is very realistic that the context of actual race has effect. This statement would support the idea of controlling for it. However, this variable takes MANY different values, so including it would introduce a lot of dummy variables. It would be not only unpractical, but could also introduce computational issues. One solution would be using a hierarchical (i.e. multilevel) model. But since we did not cover this in our course, we will simply **ignore** this variable. If you are interested, we can briefly dicuss the rationale behind hierarchical models in class.
* *location* - it is very well possible that location plays some role, however, there are three different types of locations (state, district, national) and plenty of different specific locations (specific states and districts) in this variable. We think we cover some of the interesting variation already with the variable type\_simple (type of elections). It could still make sense to try to include location in a multilevel model, but again, we are **not** doing this in this course.
* *type\_detail* - if someone were interested in detailed analysis of this, it could make sense to include this instead of type\_simple, but it has many categories, so many dummy variables would be produced. Probably only useful for specific goals, we are **not** including this one.
* *pollster* - there are plenty different pollsters, it could work in a multilevel or after some merging (if there is a good key to merge some pollsters together). For our analysis, we **disregard** this variable.
* all the \*"\_party"\* variables - it could actually be an interesting research question to see how the errors differ in relation to whether a Republic or a Democrat or a member of other party is the leading runner, or based on how close the race is. However, these are all different research questions than ours. You can surely develop arguments to also control for these in our research design, but we decided to **disregard** these considerations from our example

*You should NOT include the following:*

* *rightcall* - you should NOT include this variable in the model because it is a potential collider - whether or not the poll called the outcome correctly is likely influenced by the method (provided that any of the three methods really is better or worse, but we assume it is possible or we would not run this analysis). Yet it is also likely influenced by the error (polls with smaller errors are expected to call the outcome correctly more often). When both the dependent and the main independent variable may be reasonably argued to cause a third variable, this third variable should not be included in the model (should not be controlled for).
* *bias* - you should also NOT include this variable in the model for similar reasons as rightcall. Again, polling method can likely influence bias. And bias can influence the value of error - indeed, error is computationally directly derived from the the bias. When a third variable is an intermediate between the main independent variable and the dependent variable, we should not control for it.

*Also not to be included* - *pollno* - only a technical variable without substantive meaning - all the \*"\_name"\* variables - unimportant for the goal of our analysis - all the \*"\_pct"\* variables - all we need from them is already carried by the variable *error*, so we can disregard them.

These latter two variables could be interesting for other research questions, though. We encourage you to think what they could be (example is there more error when Democractic / Republican candidate is leading the race etc.).

### Comments related to your submitted assignments

Most of you did not have problems with this part. The key thing in this part is to correctly recognize the nature of the model and select controlling variables correspondingly. The goal of our analysis is to estimate the effect of sampling methodology on the prediction error of the pre-election polls. Therefore, we should control for all variables we suspect to be confounders (and avoid controlling for those we suspect may be colliders).

# STEP 3: Variables description

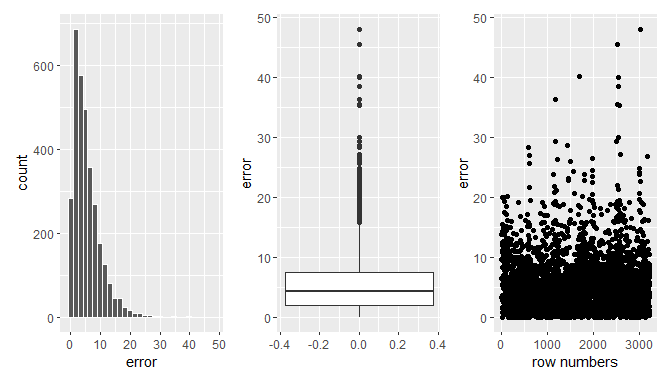
Before we start modeling, we should always pay attention to getting to know our variables. We can do either tabular or graphical description of our variables. Or both.

### Dependent variable

sum(is.na(polls$error)) # no missing values here

## [1] 0

error\_h <-   
 polls %>%   
 ggplot(aes(x=error)) +  
 geom\_histogram(color = "white")  
  
error\_b <-  
 polls %>%   
 ggplot(aes(y=error)) +  
 geom\_boxplot()  
  
error\_p <-  
 polls %>%   
 ggplot(aes(y=error, x=seq\_along(error)))+  
 geom\_point()+  
 labs(x="row numbers")  
  
error\_h + error\_b + error\_p # this patchwork functionality enables to put multiple plot together



As we see above, the *error* variable is skewed with large majority of polls having error between 0 and 10 percentage points. Some errors are extreme. Someone could consider leaving out the cases with extreme errors. That can be fine, but we will proceed with all observations (assuming extreme errors can happen as part of life rather than being glitches in the data).

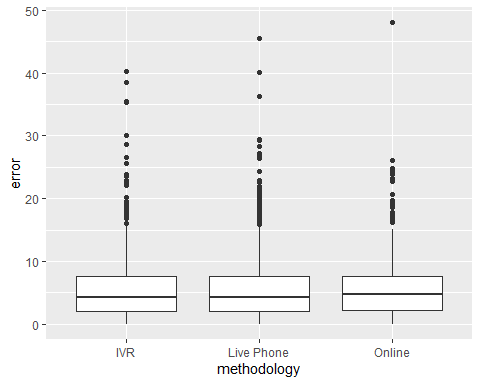
### Main independent variable

In the outputs below, we first look at the frequencies of methodology types in out data. We then look both visually and analytically at the bivariate relationship between the dependent and the main independent variable (i.e., without controlling for anything). There does not seem to be any significant difference among methodologies in the resulting error. However, this is no reason to stop the analysis. Some differences may be “hidden” due to confounders. So we still need to conduct the full analysis with controls to make sure.

polls %>% count(methodology) # no missing values here

## # A tibble: 3 x 2  
## methodology n  
## <chr> <int>  
## 1 IVR 844  
## 2 Live Phone 1876  
## 3 Online 511

# first, we can look at the raw differences of errors across different methodologies without controlling for anything...  
  
# ... visually  
  
polls %>% ggplot(aes(x=methodology, y=error))+  
 geom\_boxplot()



# ... or analytically  
  
m1 <- lm(error ~ methodology, data = polls)  
summary(m1, digits=3)

##   
## Call:  
## lm(formula = error ~ methodology, data = polls)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.613 -3.463 -1.153 2.053 42.427   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.510438 0.166886 33.019 <0.0000000000000002 \*\*\*  
## methodologyLive Phone -0.007389 0.200950 -0.037 0.971   
## methodologyOnline 0.122556 0.271757 0.451 0.652   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.848 on 3228 degrees of freedom  
## Multiple R-squared: 9.279e-05, Adjusted R-squared: -0.0005267   
## F-statistic: 0.1498 on 2 and 3228 DF, p-value: 0.8609

### Other independent variables

Before we look at the individual independent variables in turns, we will check missingness for all of them at one go. This is where your dplyr skills come in handy.

polls %>% select(year, type\_simple, polldate, electiondate, samplesize, partisan) %>%   
 summarise(across(.cols = everything(),  
 .fns = ~sum(is.na(.))))

## # A tibble: 1 x 6  
## year type\_simple polldate electiondate samplesize partisan  
## <int> <int> <int> <int> <int> <int>  
## 1 0 0 0 0 0 0

Oh, this is such non problematic data! No missing values! Thank you, Ales, for finding this dataset.

Next, if we have time (which we should if we care about the analysis), it is best to start with understanding both univariate distribution of each independent variable (it helps us detect where we can expect problems) and bivariate distribution against the dependent variable (it helps as predict which of the variables we control for are likely to have big effect in the final model). However, the bivariate analysis is only indicative, it is no substitute for diagnostics of the final model.

We think the best way to do this is visually. All the code to produce the visual analysis follows. Notice that the variable time\_gap is not part of the dataset, it needs to be constructed.

# year variable  
  
year\_uni <-  
 polls %>% ggplot(aes(x=year))+  
 geom\_bar()  
  
year\_bi <-  
 polls %>% ggplot(aes(x=year, y=error))+  
 geom\_point()+  
 geom\_smooth()  
  
# type\_simple  
  
type\_uni <-  
 polls %>% ggplot(aes(x=type\_simple))+  
 geom\_bar() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))  
  
type\_bi <-   
 polls %>% ggplot(aes(x=type\_simple, y=error))+  
 geom\_boxplot()+  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))  
  
# time\_gap - we first need to calculate time gap between the polls and the actual elections   
  
polls$polldate %>% str() # the variables are character, not date, so we have to transform them before we can subtract them from each other

## chr [1:3231] "10/13/1998" "10/13/1998" "10/13/1998" "10/14/1998" ...

# first, always check the transformations does what you want  
polls %>%   
 mutate(polldate\_d = as.Date(polldate, format = "%m/%d/%Y"),  
 electiondate\_d = as.Date(electiondate, format = "%m/%d/%Y")) %>%   
 select(polldate, polldate\_d, electiondate, electiondate\_d) %>%   
 head()

## # A tibble: 6 x 4  
## polldate polldate\_d electiondate electiondate\_d  
## <chr> <date> <chr> <date>   
## 1 10/13/1998 1998-10-13 11/3/1998 1998-11-03   
## 2 10/13/1998 1998-10-13 11/3/1998 1998-11-03   
## 3 10/13/1998 1998-10-13 11/3/1998 1998-11-03   
## 4 10/14/1998 1998-10-14 11/3/1998 1998-11-03   
## 5 10/14/1998 1998-10-14 11/3/1998 1998-11-03   
## 6 10/14/1998 1998-10-14 11/3/1998 1998-11-03

# only then proceed with the transformation   
  
polls <-   
 polls %>%   
 mutate(polldate\_d = as.Date(polldate, format = "%m/%d/%Y"),  
 electiondate\_d = as.Date(electiondate, format = "%m/%d/%Y"),  
 time\_gap = electiondate\_d - polldate\_d,  
 time\_gap = as.numeric(time\_gap))  
  
time\_gap\_uni <-  
 polls %>% ggplot(aes(y=time\_gap, x=""))+  
 geom\_boxplot()+  
 labs(x = "")  
  
time\_gap\_bi <-  
 polls %>%   
 ggplot(aes(x=time\_gap, y=error))+  
 geom\_point()+  
 geom\_smooth()  
  
# samplesize  
  
# oh, there is extreme skew here...  
samplesize\_uni\_raw <-  
 polls %>% ggplot(aes(y=samplesize, x=""))+  
 geom\_boxplot()+  
 labs(x = "")  
  
# we shoudl consider some transformation such log  
samplesize\_uni\_log <-  
 polls %>% ggplot(aes(y=log10(samplesize),x=""))+  
 geom\_boxplot()+  
 labs(x = "")  
  
samplesize\_bi <-  
 polls %>% ggplot(aes(x=log10(samplesize), y=error))+  
 geom\_point()+  
 geom\_smooth()  
  
  
# partisan   
  
partisan\_uni <-  
 polls %>% ggplot(aes(x=partisan))+  
 geom\_bar()  
  
partisan\_bi <-  
 polls %>% ggplot(aes(x=partisan, y = error))+  
 geom\_boxplot()

The plot composition below always look at one independent variable in one row. Normally, there is the univariate distribution to the left and bivariate to the right. For sample size, there is also the log-tranformed graph as the distribution is otherwise too extremely skewed. The bivarite graph for sample size also uses this log-transformed sample size.

For the *year* variable, it seems that the number of polls per year very much depends on how many and how important elections took place each year, i.e. the election cycle (plot A). However, there does not seem to be any eye-catching relationship between what year the polls took place and what the error was in terms of significant reduction or increase in error over time (plot B).

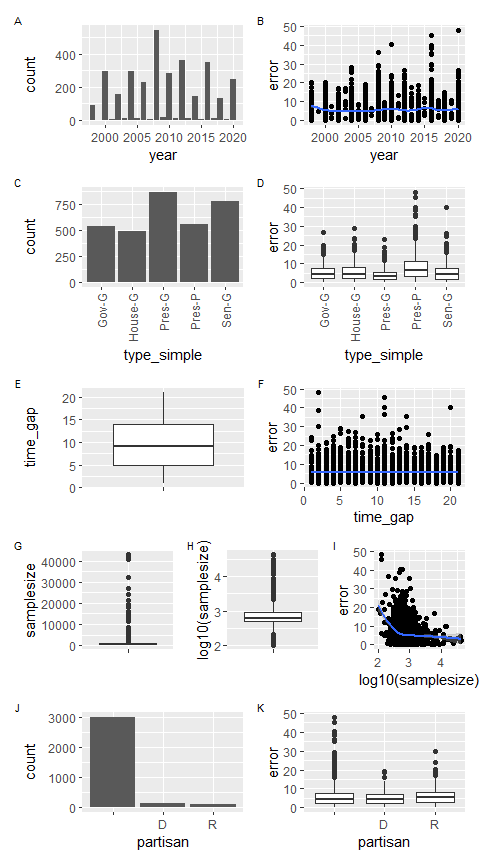
There is between ca 450 and 850 observations for each election type (considering the five types we decided to refer to, plot C). The type of elections seems to be associated with the error at least to some extent, for example, Presidential primaries tend to have the largest error, whereas Presidential general elections the smallest (plot D).

Noteworthy, time gap in all the polls in the sample is small, between 0 and 20 days or so. In addition, there seems to be no big relationship between the time gap between the polls and actual elections on the one hand and the error on the other hand (F)

Sample size unsurprisingly plays some role, but in accordance with statistical theory, the relationship with error is not linear (plot I). The error is significantly bigger for very small samples, but goes down quickly and levels off. This could be a hint that we should use some tool to model non-linearity, but not necessarily, we cannot know ahead what will happen in the multidimensional world of multiple linear regression.

Partisan polls are rare (plot J) and their effect on error seems to be small if any. The error for Republican commissioned polls seems slightly higher than the rest of the polls (plot K).

(year\_uni + year\_bi) / (type\_uni + type\_bi) / (time\_gap\_uni + time\_gap\_bi) /   
 (samplesize\_uni\_raw + samplesize\_uni\_log + samplesize\_bi) /  
 (partisan\_uni + partisan\_bi) +  
 plot\_annotation(tag\_levels = 'A') &   
 theme(plot.tag = element\_text(size = 8))



### Comments related to your submitted assignments

Again, most of you did not have problems with this part. You correctly plotted relevant variables, sometimes grouped using the methodology variable. For a heavily skewed variables, such as the sample size, it may be beneficial to use an appropriate transformation (such as taking the log).

# STEP 4: Initial model presentation and interpretation

First, we look at the model when we just let all independent variables in the model as untransformed main effects. We suspect this could be problematic (we saw the huge skew of sample size and the non-linear relationship between error and sample size), but we are going to use diagnostics to check if any serious problems appear, anyway.

Looking at summary, we notice some statistically significant relationships (other than statistically significant differences can probably be disregarded straight away as we have a really big sample).

First and most importantly, Online method of polling produces 0.79 (0.8) percentage point greater errors than Interactive voice response (IVR, the reference category). Since we work within framework of multiple linear regression, we should stress what this means exactly: when comparing the errors of polls which are the same in terms of year, election type, the time gap between the polls and the elections, the sample size and the consideration whether or not the poll was commissioned as partisan, Online methodology produce errors bigger by 0.8 percentage points than IVR.

Life phone produces slightly smaller error (by 0.029 percentage point) than the IVR methodology, however the difference is negligible (and also not statistically significant). The conclusion is that polls comparable in all the other variables produce about the same error whether they are life phone interviews or IVRs.

We actually do not need to agonize too much about the other coefficients in the tables, they are just controls instrumental for answering our main research question. Yet it may be interesting to notice that there is relatively bigger error in the Presidental primaries and, in contrast, smaller error in Presidential general elections. Time gap, does, in fact, increases the error (it did not look so in the bivariate analysis), other things in the model being equal: polls taken 1 day further away from the election produce 0.06 percentage points worse estimates than comparable polls taken 1 day closer to the elections. In other words, 10 days difference correspondes to 0.6 percentage point difference in error (within the period we look at, this is, of course, something we should not be tempted to generalize beyond - it is unlikely that polls 1000 days from the election will have 60 % error on average). Sample size decreases error. Republican-commissioned polls display larger errors.

m2 <- lm(error ~ methodology + year + type\_simple + time\_gap + samplesize + partisan, data = polls)  
  
print(summary(m2), digits=1)

##   
## Call:  
## lm(formula = error ~ methodology + year + type\_simple + time\_gap +   
## samplesize + partisan, data = polls)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9 -3 -1 2 39   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -11.68201 30.19427 -0.4 0.699   
## methodologyLive Phone -0.02952 0.19441 -0.2 0.879   
## methodologyOnline 0.79637 0.28125 2.8 0.005 \*\*   
## year 0.00806 0.01504 0.5 0.592   
## type\_simpleHouse-G 0.43297 0.29218 1.5 0.138   
## type\_simplePres-G -1.11897 0.25659 -4.4 0.00001 \*\*\*  
## type\_simplePres-P 3.29191 0.28889 11.4 < 0.0000000000000002 \*\*\*  
## type\_simpleSen-G 0.02181 0.26011 0.1 0.933   
## time\_gap 0.06411 0.01541 4.2 0.00003 \*\*\*  
## samplesize -0.00012 0.00004 -2.6 0.009 \*\*   
## partisanD 0.12297 0.43108 0.3 0.775   
## partisanR 1.00230 0.45639 2.2 0.028 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5 on 3219 degrees of freedom  
## Multiple R-squared: 0.09, Adjusted R-squared: 0.09   
## F-statistic: 3e+01 on 11 and 3219 DF, p-value: <0.0000000000000002

# STEP 5: Initial model fit and diagnostics

We already could assess model fit in the summary above. With R-squared of 0.09, we know the predictive power of the model is modest, but not negligible. Knowing the values of all variables in the model, we predict 9 % of the variance in error across the models.

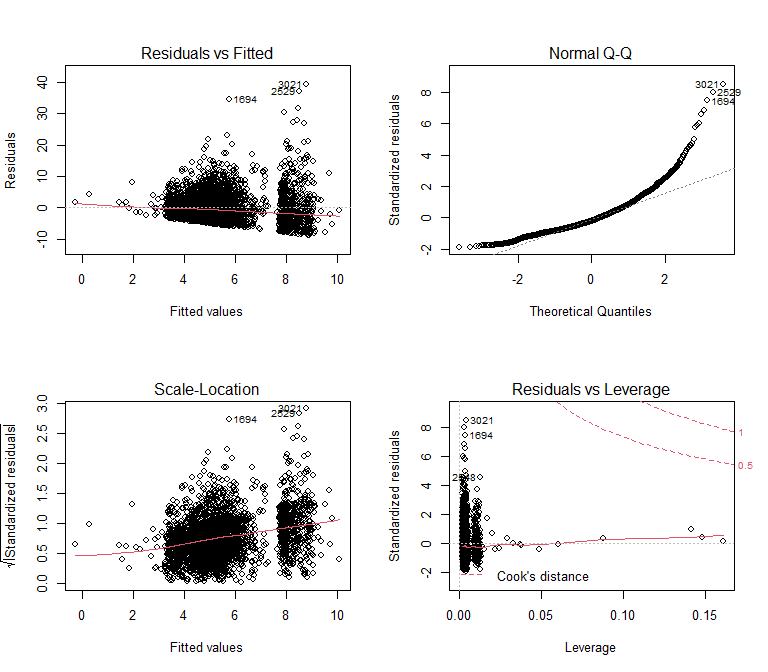
Next, we should see the diagnostic plot.

Maybe first remember the assumptions we make:

* validity - as we saw in the variables selection section, there are some variables theoretically interesting (what if the pollster differ in quality and also their use of methodology? In multiple linear model, we are actual assuming that all the polls are independent, but they are not - they are clustered in the pollster or agencies. And also location.). However, we would need multilevel model to address this. Also the validity of the variable *partisan* is questioned by the authors of the data set. These things are boyond what we can tackle here, but we can still inquire in the model specification (also a matter of validity) using some diagnostic graphs.
* representativeness - we did not dig into this, but we believe FiveThirtyEight have thought about it and the polls are a good image of the polling landscape in the US. It is probably sensible to look at the polls as a census of reasonable polls out there rather than a sample, but we actually did not go into detail about this. It is something we would have to investigate if we wanted to publish this analysis.
* linearity - We will see using diagnostic plots to see if the relationship in the model to see if the residuals average around 0 for different levels of the predicted values of the dependent variable
* independence of errors - well, like we said, we assume each poll is independent when, in fact, there is a limited number of polster, and *location* contexts … so maybe the errors are not quite independent. We don’t expect this to be a horrible issue here, but a multilevel model accounting for this could be better here.
* homoscedasticity - we will see in the diagnotic plots
* normality of errors - we will see in the diagnotic plots

Now, let’s see the diagnostic plots:

par(mfrow = c(2,2)) #adjusts how plot are displayed in the plots pane of R  
# you can reset the above to default by either restarting R Studio or running par(mfrow = c(1,1))  
plot(m2)



par(mfrow = c(1,1))

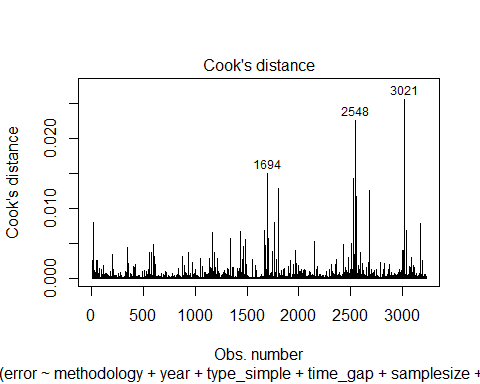
How can we interpret them?

* the first (residuals vs fitted) is mainly used to assess linearity of the modeled relationships (can we reasonable model the dependent variable as a linear combination of our predictors?) - since the red line does not deviate much from the dotted and is straight rather than wiggly, we can say that our specification of the model does not violate the linearity assumption too much. However, we can also notice the discontinuity in the data - there is (almost) no fitted values of 7.5 - 8. This may look weird, but it can happen as we use categorical predictors. It is probably not a reason for serious concerns.
* the second (normal Q-Q) is used to assess the normality of residuals. We see some serious departure from normality (points should be around the dotted line if the assumption holds), so this assumption is violated. However, remember that this assumption is actually comparatively least important and is only relevant if we need to make predictions on individual level (for individual polls) or when we use small sample sizes. In case we are interested in predicting individual cases, we could try to make separate groups for different sample sizes or some transformations until we are satisfied. Since we are not interested in such exercise here, we will just let it be.
* the third (scale-location) is used to assess the assumption of homoscedasticity. Is the variance of residuals similar around across different levels of fitted values? This is a bit hard to say as the number of observations is very different across different levels of fitted values. Still it looks to us that there is or could be some funneling out of the errors indicating heteroscedasticity. Practically, this means that the p-values for the coefficients may be biased (i.e. incorrect p-values, incorrect confidence intervals). (If you want to, you can learn about robust standard errors and use them. We did not cover robust standard errors in the classes, but we provide some material on the course web page. However, since the sample size is large, we are not much worried about probably just a small inaccuracy in p-values.)
* the forth (residuals vs leverage) shows values with large Cook’s distance. With our sample size, it is quite unlikely we would see any significant distortion of our model estimated due to a few influential cases, but we can still have a look at those which exert most influence. Anyway, the good news is that the cases with big leverage are close to the dotted line - do not exert much influence on the model estimates.

### After diagnostics - Cook’s distance

After looking at the “global” diagnostic charts, we can look at individual most influential cases. We can use the plot which identifies the three most influential and focus on them. There are many ways to do it, we will use *rownames\_to\_column* function to transform rownames to a variable which we can further use for filtering. When we tabulate the three cases (only variables used in the model - see the select command below), we see they are all polls with extreme errors, two of them presidential primaries. We will let it be at this point (we are not interested in individual observations in our exercise), but you could also consider removing these (and perhaps other) cases for some other purposes.

plot(m2, which =4)



polls <- rownames\_to\_column(polls, var = "rownames")  
  
polls %>%   
 filter(rownames %in% c(1694,2548,3021)) %>%   
 as\_tibble() %>%   
 select(error, methodology, year, type\_simple, time\_gap, samplesize, partisan)

## # A tibble: 3 x 7  
## error methodology year type\_simple time\_gap samplesize partisan  
## <dbl> <chr> <int> <chr> <dbl> <dbl> <chr>   
## 1 40.2 IVR 2010 Sen-G 20 500 ""   
## 2 30.0 IVR 2016 Pres-P 2 1060 "R"   
## 3 48.1 Online 2020 Pres-P 2 125 ""

If you need more flexibility and focus on your-specified number of cases, not just three shown in the plot, you can do the following.

polls$cd <- cooks.distance(m2) # save cook's distance in the original data  
  
 # display all observations with cook's distance bigger than given value  
polls %>% filter(cd > 0.01) %>%   
 select(error, methodology, year, type\_simple, time\_gap, samplesize, partisan, cd)

## # A tibble: 8 x 8  
## error methodology year type\_simple time\_gap samplesize partisan cd  
## <dbl> <chr> <int> <chr> <dbl> <dbl> <chr> <dbl>  
## 1 40.2 IVR 2010 Sen-G 20 500 "" 0.0150  
## 2 23.8 IVR 2010 House-G 12 461 "R" 0.0128  
## 3 45.5 Live Phone 2016 Pres-P 11 123 "" 0.0143  
## 4 30.0 IVR 2016 Pres-P 2 1060 "R" 0.0226  
## 5 40.0 Live Phone 2016 Pres-P 11 600 "" 0.0104  
## 6 38.4 IVR 2016 Pres-P 2 475 "" 0.0117  
## 7 4.4 Online 2016 Pres-G 12 40816 "" 0.0126  
## 8 48.1 Online 2020 Pres-P 2 125 "" 0.0256

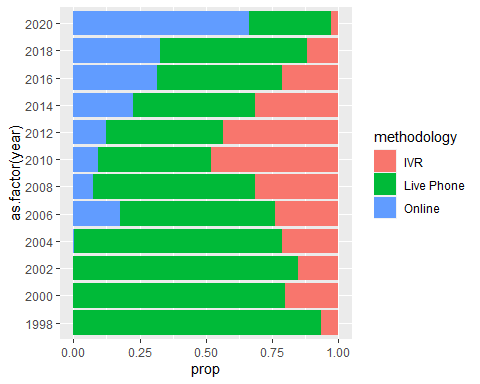
# display all observations with cook's distance bigger than given quantile, in this case 99.9%  
polls %>% filter(cd > quantile(cd, 0.999)) %>%   
 select(error, methodology, year, type\_simple, time\_gap, samplesize, partisan, cd)

## # A tibble: 4 x 8  
## error methodology year type\_simple time\_gap samplesize partisan cd  
## <dbl> <chr> <int> <chr> <dbl> <dbl> <chr> <dbl>  
## 1 40.2 IVR 2010 Sen-G 20 500 "" 0.0150  
## 2 45.5 Live Phone 2016 Pres-P 11 123 "" 0.0143  
## 3 30.0 IVR 2016 Pres-P 2 1060 "R" 0.0226  
## 4 48.1 Online 2020 Pres-P 2 125 "" 0.0256

# STEP 6: Adjusting our model

We can further try to make adjustments to our model specification. For example, we saw that the extreme skew of sample size could be causing problems. We could use log of sample size. One student in the class also noticed that the proportion of methodology over time was not constant. Very much on the contrary. Maybe, we should try to see if there is interaction between the two.

# First plot methodology by year (only for years with 50+ polls) to see the difference. Maybe we could make live phone the reference category as it is most stable across the years.  
polls %>% group\_by(year) %>% mutate(year\_n = length(year)) %>% filter(year\_n > 50) %>%   
 group\_by(year, methodology) %>%   
 summarise(n = n()) %>%   
 mutate(prop = n/sum(n)) %>%   
 ggplot(aes(y=as.factor(year), x=prop, fill=methodology))+  
 geom\_col(position = "fill")



# Changing reference category in methodology.   
  
polls <-  
 polls %>% mutate(methodology = fct\_relevel(methodology, "Live Phone")) # set as first level, i.e. reference category  
  
  
# We could also consider centering as we want to use interactions. In that case, centering helps to interpret the regression tables, specifically the main effects. However, since we are going to use plots of marginal effects for model interpretation, it would not be strictly necessary here (the required transformations also happen on the background of plotting marginal effects). Yet for the sake of exercise, we show how centering is done and you can look at how it influences the regression coefficients. Note that we have to log first and then center. The other order would not work - cannot log negative value.  
  
polls <-  
 polls %>% mutate(  
 year\_c = year - mean(year),  
 time\_gap\_c = time\_gap - mean(time\_gap),  
 samplesize\_log\_c = log10(samplesize) - mean(log10(samplesize)))

# Now, run the newly specified model with raw and with centered variables (for the sake of illustration).   
  
m3 <- lm(error ~ methodology\*year + type\_simple + time\_gap + log10(samplesize) + partisan, data = polls)  
summary(m3)

##   
## Call:  
## lm(formula = error ~ methodology \* year + type\_simple + time\_gap +   
## log10(samplesize) + partisan, data = polls)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.111 -3.049 -0.915 2.103 37.036   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 66.57020 35.55484 1.872 0.061252 .   
## methodologyIVR -295.28686 78.10648 -3.781 0.000159 \*\*\*  
## methodologyOnline -254.00311 96.35819 -2.636 0.008429 \*\*   
## year -0.02733 0.01769 -1.545 0.122533   
## type\_simpleHouse-G 0.19426 0.29237 0.664 0.506466   
## type\_simplePres-G -1.07129 0.25608 -4.183 0.00002949160624 \*\*\*  
## type\_simplePres-P 2.94085 0.29149 10.089 < 0.0000000000000002 \*\*\*  
## type\_simpleSen-G 0.02777 0.25829 0.108 0.914395   
## time\_gap 0.06087 0.01532 3.974 0.00007231078010 \*\*\*  
## log10(samplesize) -2.57015 0.37231 -6.903 0.00000000000609 \*\*\*  
## partisanD -0.06240 0.42864 -0.146 0.884263   
## partisanR 0.78185 0.45483 1.719 0.085711 .   
## methodologyIVR:year 0.14711 0.03888 3.784 0.000157 \*\*\*  
## methodologyOnline:year 0.12672 0.04785 2.649 0.008121 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.596 on 3217 degrees of freedom  
## Multiple R-squared: 0.1045, Adjusted R-squared: 0.1009   
## F-statistic: 28.89 on 13 and 3217 DF, p-value: < 0.00000000000000022

m3\_c <- lm(error ~ methodology\*year\_c + type\_simple + time\_gap\_c + samplesize\_log\_c + partisan, data = polls)  
summary(m3\_c)

##   
## Call:  
## lm(formula = error ~ methodology \* year\_c + type\_simple + time\_gap\_c +   
## samplesize\_log\_c + partisan, data = polls)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.111 -3.049 -0.915 2.103 37.036   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.95114 0.21598 22.924 < 0.0000000000000002 \*\*\*  
## methodologyIVR 0.30834 0.19678 1.567 0.117230   
## methodologyOnline 0.63038 0.35034 1.799 0.072062 .   
## year\_c -0.02733 0.01769 -1.545 0.122533   
## type\_simpleHouse-G 0.19426 0.29237 0.664 0.506466   
## type\_simplePres-G -1.07129 0.25608 -4.183 0.00002949160624 \*\*\*  
## type\_simplePres-P 2.94085 0.29149 10.089 < 0.0000000000000002 \*\*\*  
## type\_simpleSen-G 0.02777 0.25829 0.108 0.914395   
## time\_gap\_c 0.06087 0.01532 3.974 0.00007231078010 \*\*\*  
## samplesize\_log\_c -2.57015 0.37231 -6.903 0.00000000000609 \*\*\*  
## partisanD -0.06240 0.42864 -0.146 0.884263   
## partisanR 0.78185 0.45483 1.719 0.085711 .   
## methodologyIVR:year\_c 0.14711 0.03888 3.784 0.000157 \*\*\*  
## methodologyOnline:year\_c 0.12672 0.04785 2.649 0.008121 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.596 on 3217 degrees of freedom  
## Multiple R-squared: 0.1045, Adjusted R-squared: 0.1009   
## F-statistic: 28.89 on 13 and 3217 DF, p-value: < 0.00000000000000022

# STEP 7: Comparing our initial and new model

Since we now have two candidate models for final interpretation, we should compare them formally.

anova(m2, m3\_c)

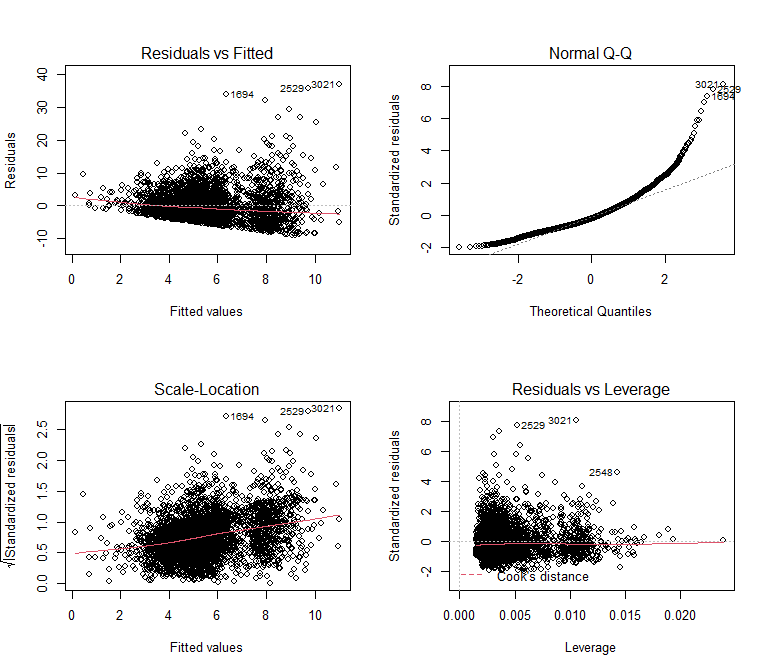
## Analysis of Variance Table  
##   
## Model 1: error ~ methodology + year + type\_simple + time\_gap + samplesize +   
## partisan  
## Model 2: error ~ methodology \* year\_c + type\_simple + time\_gap\_c + samplesize\_log\_c +   
## partisan  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 3219 69026   
## 2 3217 67953 2 1073 25.398 0.00000000001137 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

We see that the latter (more complicated) model has resulted in decrease of RSS by a statistically significant amount. It better fit does not seem to be caused simply by chance.

# STEP 7: New model diagnostics

Similar to the initial model, the same limitations / interpretations apply.

par(mfrow = c(2,2)) #adjusts how plot are displayed in the plots pane of R  
# you can reset the above to default by either restarting R Studio or running par(mfrow = c(1,1))  
plot(m3\_c)

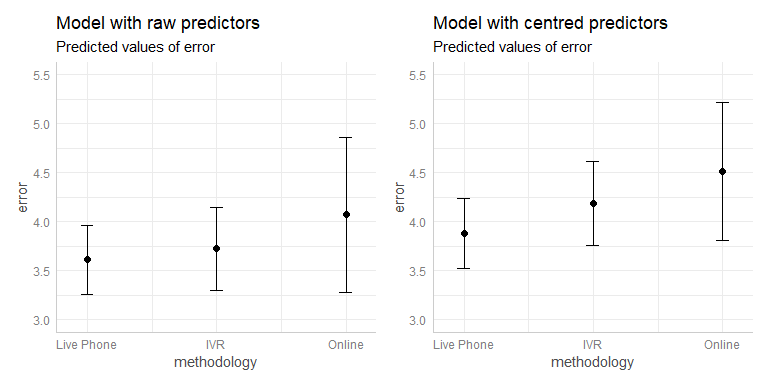


par(mfrow = c(1,1))

# STEP 8: Final model presentation and interpretation

First, we just look at marginal effects of different methodologies. Remember, marginal effects are expected values of the dependent variable for given values of selected independent variable while other independent variables are held constant.

main\_effect\_raw <- plot(ggpredict(m3, terms = c("methodology"))) + scale\_y\_continuous(limits = c(3,5.5)) + labs(title = "Model with raw predictors", subtitle = "Predicted values of error")  
main\_effect\_c <- plot(ggpredict(m3\_c, terms = c("methodology"))) + scale\_y\_continuous(limits = c(3,5.5)) + labs(title = "Model with centred predictors", subtitle = "Predicted values of error")  
  
main\_effect\_raw + main\_effect\_c



If we plot marginal effects for both the model with centered and the model with raw variables, we notice a different result (the two plots above are different). Why is that? The answer is in the background transformations when ggpredict “holds other variables constant”. There are actually two issues. The first is that year and year\_c are saved as different kind of vector. While year is integer, it necessarily becomes “double” after centering (allowing for decimals). Ggpredict treats integers differently - it holds contant at its modus, in this case year 2008. Whereas it holds contant at the mean of a “double”, see the “adjusted for” notes below. The second issue is with the log transformation. In model 3 (m3), we conducted the transformation in specifying the model, so ggpredict is able to undo the transformation and hold constant at the mean of the original variable. In contrast, the centred model (m3\_c) uses a variable created elsewhere (samplesize\_log\_c), so ggpredict takes the mean of it, i.e. the mean of log, which is different from taking the mean of the orginal raw variables. For illustration purposes, see below how we could get the same result. Such small things can be pain when working, so it is good to kow about them and learn to estimate where problems could be happening.

# year and year\_c are different   
polls %>% select(year, year\_c) %>% glimpse()

## Rows: 3,231  
## Columns: 2  
## $ year <int> 1998, 1998, 1998, 1998, 1998, 1998, 1998, 1998, 1998, 1998, ...  
## $ year\_c <dbl> -11.34602, -11.34602, -11.34602, -11.34602, -11.34602, -11.3...

# the the plots are based on different numbers, we can look at them here  
ggpredict(m3, terms = c("methodology"))

## # Predicted values of error  
## # x = methodology  
##   
## x | Predicted | 95% CI  
## -------------------------------------  
## Live Phone | 3.61 | [3.26, 3.97]  
## IVR | 3.72 | [3.30, 4.15]  
## Online | 4.07 | [3.28, 4.86]  
##   
## Adjusted for:  
## \* year = 2008.00  
## \* type\_simple = Pres-G  
## \* time\_gap = 9.94  
## \* samplesize = 922.43  
## \* partisan =

ggpredict(m3\_c, terms = c("methodology"))

## # Predicted values of error  
## # x = methodology  
##   
## x | Predicted | 95% CI  
## -------------------------------------  
## Live Phone | 3.88 | [3.53, 4.23]  
## IVR | 4.19 | [3.76, 4.62]  
## Online | 4.51 | [3.81, 5.21]  
##   
## Adjusted for:  
## \* year\_c = 0.00  
## \* type\_simple = Pres-G  
## \* time\_gap\_c = 0.00  
## \* samplesize\_log\_c = -0.00  
## \* partisan =

# we could recalculate model 3 so that the results after processing with ggpredict are the same  
polls <-  
 polls %>% mutate(  
 samplesize\_log = log10(samplesize))  
  
m3.1 <- lm(error ~ methodology\*as.numeric(year) + type\_simple + time\_gap + samplesize\_log + partisan, data = polls)  
main\_effect\_raw <- plot(ggpredict(m3.1, terms = c("methodology"))) + scale\_y\_continuous(limits = c(3,5.5)) + labs(title = "Model with raw predictors", subtitle = "Predicted values of error")  
  
ggpredict(m3.1, terms = c("methodology"))

## # Predicted values of error  
## # x = methodology  
##   
## x | Predicted | 95% CI  
## -------------------------------------  
## Live Phone | 3.88 | [3.53, 4.23]  
## IVR | 4.19 | [3.76, 4.62]  
## Online | 4.51 | [3.81, 5.21]  
##   
## Adjusted for:  
## \* year = 2009.35  
## \* type\_simple = Pres-G  
## \* time\_gap = 9.94  
## \* samplesize\_log = 2.85  
## \* partisan =

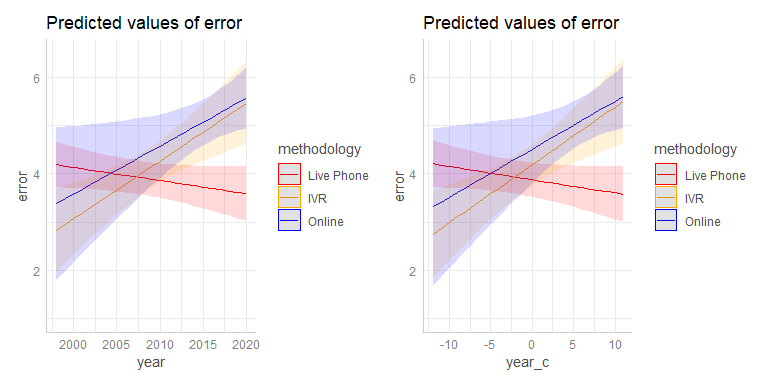
ggpredict(m3\_c, terms = c("methodology"))

## # Predicted values of error  
## # x = methodology  
##   
## x | Predicted | 95% CI  
## -------------------------------------  
## Live Phone | 3.88 | [3.53, 4.23]  
## IVR | 4.19 | [3.76, 4.62]  
## Online | 4.51 | [3.81, 5.21]  
##   
## Adjusted for:  
## \* year\_c = 0.00  
## \* type\_simple = Pres-G  
## \* time\_gap\_c = 0.00  
## \* samplesize\_log\_c = -0.00  
## \* partisan =

Back to model interpretation. To fully use the information carried by the model, we want to show the effect of the significant interaction. We do so by plotting a *ggpredict* object with both predictors used in the interaction as *terms*.

Notice that the result is the same for both the model with raw predictors and the one with centered. Only the x axis is different - centered years are numbers around 0. For interpretability, we would pick the plot with years on x axis.

interaction\_raw <- plot(ggpredict(m3.1, terms = c("year", "methodology")), colors = c("red", "orange", "blue")) +   
 scale\_y\_continuous(limits = c(1,6.5))  
  
interaction\_c <- plot(ggpredict(m3\_c, terms = c("year\_c", "methodology")), colors = c("red", "orange", "blue"))+   
 scale\_y\_continuous(limits = c(1,6.5))  
  
interaction\_raw + interaction\_c



Alternatively, you could prefer to plot methodology on x axis (as the main independent variable) and show year with color. We get a similar conclusion - while not much different on average, Live Phone methodology seems to be getting better and the the other two seem to be getting worse. If choosing one today, Live Phone seems preferable other things being equal.

alternative\_interaction\_raw <- plot(ggpredict(m3, terms = c("methodology", "year [1998, 2002, 2006, 2010, 2014, 2018]")))  
  
alternative\_interaction\_raw

