

ASSUMPTIONS

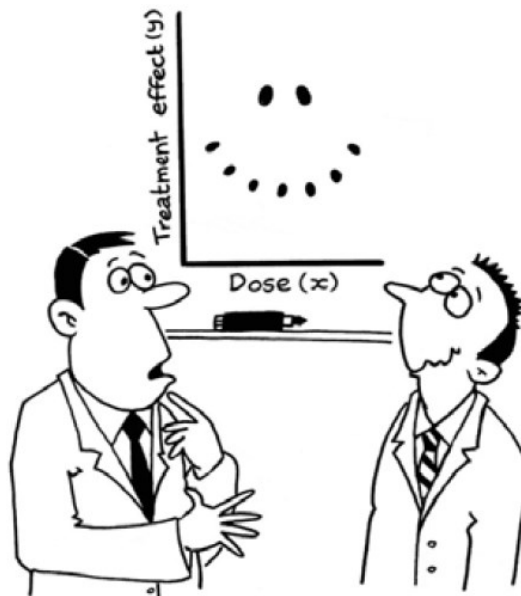
Research Methods in Psychology I & II • Department of Psychology • Colorado State University

BY THE END OF THIS UNIT YOU WILL:

1. Understand the assumptions underlying a linear regression model.
2. Know how to determine if a model meets these assumptions.
3. Learn how to detect unusual cases in your data.
4. Know how to determine if multicollinearity of the predictors presents a problem in your model.
5. Understand how remedial measures can be used to repair (or mitigate) problematic models.

Why care about assumptions?

Our work isn't complete until we check the assumptions underlying the model and evaluate the overall model fit. All models have assumptions—it's important to understand them, check them, and implement remedial actions when necessary.



"It's a non-linear pattern with outliers.....but for some reason I'm very happy with the data."

Example Dataset

Blair et al. (2004). The influence of Afrocentric facial features in criminal sentencing. *Psychological Science*, 15 (10), 674-679.

Research Question: “to determine whether...[the sentences of young Black and White male inmates] depended both on race and within race, on the degree to which they manifested Afrocentric facial features, controlling for the seriousness of the crimes they had committed and their prior criminal histories.”

Hypothesis: “controlling for legally relevant factors, Black offenders as a group may not receive harsher sentences than White offenders, but members of both groups who have relatively more Afrocentric features may receive harsher sentences than group members with less Afrocentric features.”

Study Design: From the population of all young (18 to 24 years of age) male inmates in the Florida Department of Corrections database, a sample of 216 was randomly selected, stratified by race, as designated on their court record (100 Black inmates and 116 White inmates). The researchers selected only cases involving a current offense committed between October 1, 1998, and October 1, 2002. These date restrictions ensured that the offenders in the sample were all sentenced under the same laws. Criminal histories were recorded for all inmates based on official criminal records, including the total amount of time the inmate was currently serving, the seriousness of the primary offense, the number of any additional offenses and their average seriousness, and the number of prior offenses and their average seriousness. Facial features were recorded for all inmates based on the official photo taken by the FL criminal justice system at the time of entry into prison for the current offense. The 216 facial photographs associated with the selected cases were randomly divided into two sets, each with approximately equal numbers of Black and White inmates. Each set was given to a group of undergraduate research participants ($n = 34$ and $n = 35$ respectively) who were asked to make a single, global assessment of the degree to which each face had features that are typical of African Americans, using a scale from 1 (not at all) to 9 (very much). In addition, because attractiveness and babyishness of faces have been shown to influence judicial outcomes, the participants were asked to rate the faces on these dimensions after completing the ratings for Afrocentric features.

Outcome: Sentence length in Years (years)

Predictor variable of interest: Afrocentric features (afro)

Control variables:

Severity of the primary offense (primlev)

Number of secondary offenses (nsecond)

Average severity of secondary offenses (seclev)

Number of prior offenses (nprior)

Average severity of prior offenses (priorlev)

Alternative explanation variables:

Baby-face (babyface)

Attractive (attract)

DATASET: sentence.csv

Prepare Data and Get Descriptive Statistics

Build in your SentenceReplicationNotebook in your MyClassActivities Folder.

Load libraries

```
library(tidyverse)
library(olsrr)
library(GGally)
library(psych)
library(lmtest)
library(car)
library(broom)
library(boot)
library(modelr)
```

Import data

```
sentence <- read_csv("sentence.csv")
```

Create a factor variable for race

```
sentence <- mutate(sentence, black.f = factor(black, levels = c(0,1), labels = c("White", "Black")))
```

Obtain descriptive statistics and plot data

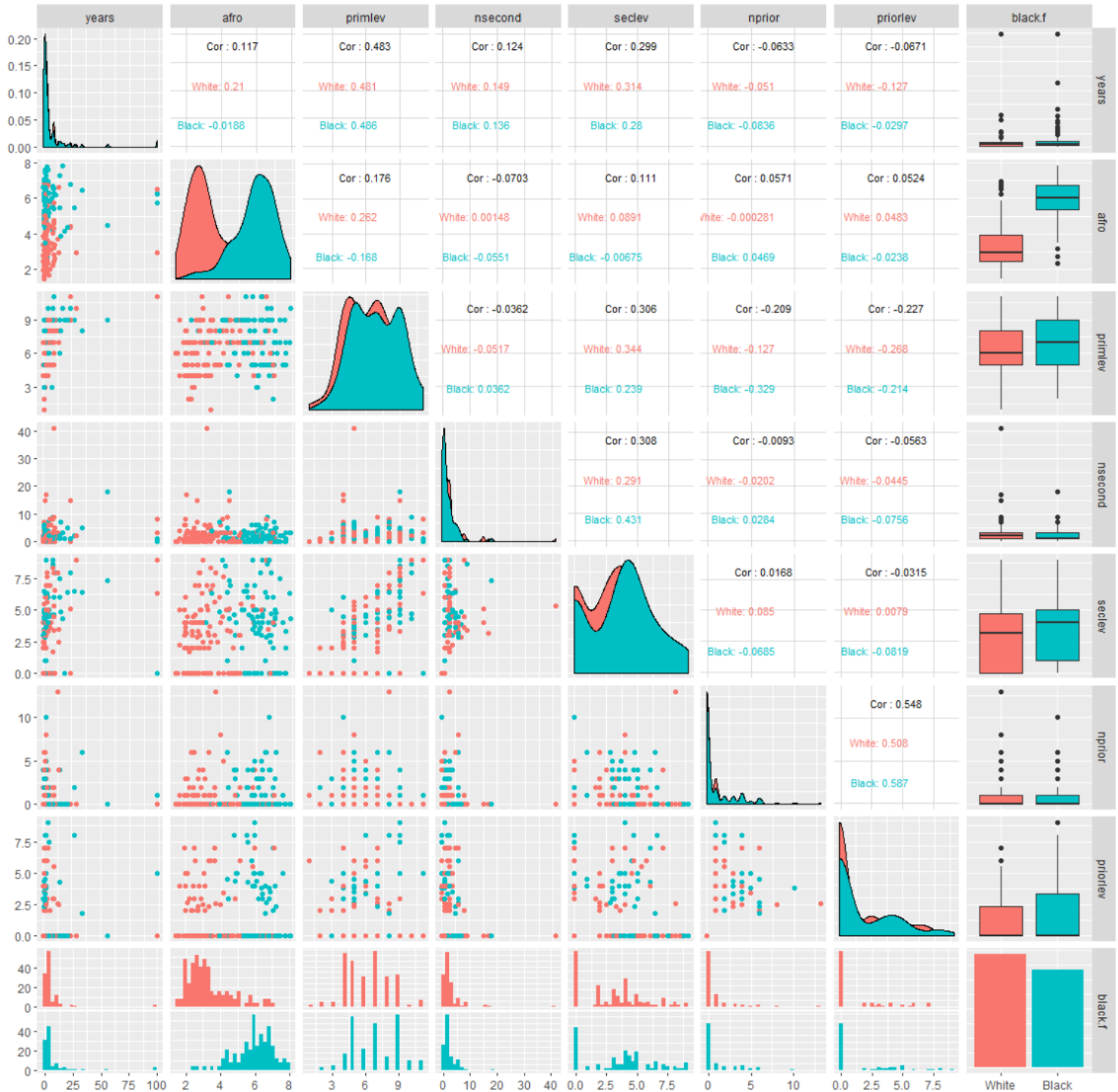
```
describe(sentence)
```

```
scatterplot <- ggpairs(sentence, columns = c("years", "afro", "primlev", "nsecond", "seclev", "nprior", "priorlev", "black.f"),
  mapping=ggplot2::aes(colour = black.f),
  upper = list(continuous = wrap("cor", size=3)),
  title = "Bivariate Relationship of Key Variables (original variables)")
print(scatterplot, progress=FALSE)
```

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
id	1	216	108.50	62.50	108.50	108.50	80.06	1.00	216.00	215.00	0.00	-1.22	4.25
years	2	216	6.84	15.54	2.67	3.50	2.11	0.42	99.00	98.58	5.02	26.18	1.06
black	3	216	0.46	0.50	0.00	0.45	0.00	0.00	1.00	1.00	0.15	-1.99	0.03
afro	4	216	4.53	1.77	4.47	4.51	2.42	1.49	7.86	6.37	0.05	-1.38	0.12
primlev	5	216	6.55	2.07	7.00	6.53	2.97	1.00	11.00	10.00	0.08	-0.67	0.14
seclev	6	216	3.40	2.57	3.65	3.24	2.44	0.00	9.00	9.00	0.20	-0.78	0.18
nsecond	7	216	2.41	3.82	1.00	1.74	1.48	0.00	41.00	41.00	5.79	49.09	0.26
anysec	8	216	0.75	0.44	1.00	0.80	0.00	0.00	1.00	1.00	-1.12	-0.75	0.03
priorlev	9	216	1.43	2.29	0.00	0.98	0.00	0.00	9.00	9.00	1.37	0.69	0.16
nprior	10	216	0.95	1.90	0.00	0.49	0.00	0.00	13.00	13.00	2.82	9.99	0.13
anyprior	11	216	0.33	0.47	0.00	0.29	0.00	0.00	1.00	1.00	0.72	-1.48	0.03
attract	12	216	3.21	0.91	3.16	3.15	0.86	1.43	6.51	5.09	0.73	0.80	0.06
babyface	13	216	4.04	1.10	3.91	4.01	1.27	1.66	7.09	5.43	0.22	-0.59	0.07
black.f*	14	216	1.46	0.50	1.00	1.45	0.00	1.00	2.00	1.00	0.15	-1.99	0.03

Scatterplot Matrix

Bivariate Relationship of Key Variables (original variables)



Prepare Variables

Create variables for MLR

```
sentence <- mutate(sentence,  
  lnyears = log(years),  
  primlev0 = primlev - 1,  
  primlev02 = primlev0^2,  
  seclev2 = seclev^2,  
  priorlev2 = priorlev^2,  
  afro_m = afro - mean(afro),  
  babyface_m = babyface - mean(babyface),  
  attract_m = attract - mean(attract))
```

Let's create and reformat all of the variables that we will need.

Blair and colleagues indicate that they took the natural log of sentence length in years because it was “positively skewed,” thus we will form a new variable called `lnyears` that is the natural log of years. All observed values of years are above 0, so taking the log works for all cases (if we had a 0, then we'd need to add some small constant to all scores and then take the log, this is because a log transformation is not defined for values less than or equal to 0). The authors of the paper also indicate that they “included quadratic terms for seriousness of the primary offense, seriousness of additional offenses, and seriousness of prior offenses, because the Florida Criminal Punishment Code specified that for more serious offenses, the length of the sentence ought to increase dramatically as the seriousness of the offense increases.” Thus, we will form squared terms of each of these three seriousness variables. While `seclev` (seriousness of secondary offenses) and `priorlev` (seriousness of prior offenses) both have an observed 0 score, there are no scores of 0 for `primlev` (seriousness of the primary offense). Therefore we will subtract a score of 1 from all `primlev` scores (1 is the lowest observed value).

Fit Model 1

Model 1: Blair and colleague's first model used multiple regression to determine the degree to which sentence length was influenced by only those factors that should lawfully predict sentencing: seriousness of the primary offense, the number and seriousness of additional concurrent offenses, and the number and seriousness of prior offenses. The authors included quadratic terms for seriousness of the primary offense, seriousness of additional offenses, and seriousness of prior offenses, because "the Florida Criminal Punishment Code specifies that for more serious offenses, the length of the sentence ought to increase dramatically as the seriousness of the offense increases." The authors also elected to log-transform sentence length due to the substantial positive skew.

Fit Model 1

```
mod1 <- lm(data=sentence, lnyears ~ primlev0 + primlev02 + nsecond + seclev + seclev2 + nprior + priorlev + priorlev2)
ols_regress(mod1)
```

Model Summary

R	0.759	RMSE	0.705
R-Squared	0.576	Coef. Var	62.473
Adj. R-Squared	0.559	MSE	0.497
Pred R-Squared	0.530	MAE	0.535

RMSE: Root Mean Square Error

MSE: Mean Square Error

MAE: Mean Absolute Error

ANOVA

	Sum of Squares	DF	Mean Square	F	Sig.
Regression	139.630	8	17.454	35.131	0.0000
Residual	102.841	207	0.497		
Total	242.471	215			

Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig.	lower	upper
(Intercept)	0.351	0.313		1.121	0.264	-0.266	0.967
primlev0	-0.137	0.115	-0.267	-1.188	0.236	-0.364	0.090
primlev02	0.039	0.010	0.875	3.790	0.000	0.019	0.059
nsecond	0.058	0.014	0.210	4.244	0.000	0.031	0.086
seclev	-0.108	0.061	-0.262	-1.770	0.078	-0.229	0.012
seclev2	0.021	0.008	0.404	2.619	0.009	0.005	0.037
nprior	0.022	0.036	0.040	0.615	0.539	-0.049	0.093
priorlev	-0.028	0.089	-0.061	-0.317	0.751	-0.203	0.147
priorlev2	0.005	0.012	0.063	0.372	0.710	-0.020	0.029

The criminal record variables accounted for a substantial amount of the variance (58%) in the log of sentence length. The seriousness of the primary offense and both the seriousness and the number of secondary offenses were significant predictors of the log of sentence length. Notice that the quadratic terms of both of these seriousness measures is positive and significant, indicating that the effect is bowl shaped, there is a ramping up of the increment to the log of years as the seriousness increases, as was expected. Neither the seriousness nor the number of prior offenses predicted the log of sentence length. The authors attribute these null effects to the "relative youthfulness of the inmates, who had relatively few prior felony offenses."

Fit Model 2

Model 2: Blair and colleague's second model added race as a predictor in order to determine if there were race differences in sentencing.

Fit Model 2

```
mod2 <- lm(data=sentence, lnyears ~ primlev0 + primlev02 + nsecond + seclev + seclev2 + nprior + priorlev + priorlev2 + black)
ols_regress(mod2)
```

Model Summary

R	0.760	RMSE	0.705
R-Squared	0.577	Coef. Var	62.506
Adj. R-Squared	0.559	MSE	0.497
Pred R-Squared	0.528	MAE	0.535

RMSE: Root Mean Square Error

MSE: Mean Square Error

MAE: Mean Absolute Error

ANOVA

	Sum of Squares	DF	Mean Square	F	Sig.
Regression	140.019	9	15.558	31.282	0.0000
Residual	102.452	206	0.497		
Total	242.471	215			

Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig.	lower	upper
(Intercept)	0.370	0.314		1.178	0.240	-0.249	0.988
primlev0	-0.135	0.115	-0.264	-1.173	0.242	-0.363	0.092
primlev02	0.039	0.010	0.879	3.804	0.000	0.019	0.059
nsecond	0.057	0.014	0.205	4.128	0.000	0.030	0.084
seclev	-0.105	0.061	-0.254	-1.710	0.089	-0.226	0.016
seclev2	0.021	0.008	0.398	2.584	0.010	0.005	0.037
nprior	0.024	0.036	0.043	0.671	0.503	-0.047	0.095
priorlev	-0.030	0.089	-0.065	-0.340	0.735	-0.205	0.145
priorlev2	0.005	0.012	0.069	0.407	0.685	-0.019	0.030
black	-0.088	0.099	-0.041	-0.884	0.377	-0.283	0.108

The race of the offender did not account for a significant amount of variance in the log of sentence length over and above the criminal record variables. The results of this analysis were “consistent with the findings of Florida’s Race Neutrality in Sentencing” report.

Fit Model 3

Model 3: In a third model, the authors added the degree to which the inmates' faces were characterized by Afrocentric features as a predictor of the log of sentence length, controlling for the race of the inmates and the criminal record.

Fit Model 3

```
mod3 <- lm(data=sentence, lnyears ~ primlev0 + primlev02 + nsecond + seclev + seclev2 + nprior + priorlev + priorlev2 + black + afro_m)
ols_regress(mod3)
```

Model Summary

R	0.767	RMSE	0.698
R-Squared	0.588	Coef. Var	61.861
Adj. R-Squared	0.568	MSE	0.487
Pred R-Squared	0.535	MAE	0.528

RMSE: Root Mean Square Error

MSE: Mean Square Error

MAE: Mean Absolute Error

ANOVA

	Sum of Squares	DF	Mean Square	F	Sig.
Regression	142.610	10	14.261	29.276	0.0000
Residual	99.861	205	0.487		
Total	242.471	215			

Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig.	lower	upper
(Intercept)	0.506	0.316		1.602	0.111	-0.117	1.129
primlev0	-0.138	0.114	-0.270	-1.211	0.227	-0.363	0.087
primlev02	0.039	0.010	0.876	3.831	0.000	0.019	0.059
nsecond	0.058	0.014	0.208	4.233	0.000	0.031	0.085
seclev	-0.110	0.061	-0.267	-1.812	0.071	-0.230	0.010
seclev2	0.021	0.008	0.409	2.676	0.008	0.006	0.037
nprior	0.021	0.036	0.037	0.588	0.557	-0.049	0.091
priorlev	-0.024	0.088	-0.051	-0.269	0.788	-0.197	0.150
priorlev2	0.004	0.012	0.055	0.325	0.746	-0.020	0.028
black	-0.322	0.141	-0.151	-2.280	0.024	-0.600	-0.044
afro_m	0.092	0.040	0.153	2.306	0.022	0.013	0.171

Afrocentric features was a significant predictor of log of sentence length over and above the effects of the other variables. Note that with Afrocentric features in the model, race becomes a significant predictor of the log of sentence length, but in the direction opposite to what one might expect — with White inmates serving longer sentences than Black inmates.

Fit Model 4

Model 4: Finally, Blair and colleagues added the alternative explanation variables — baby face and attractiveness.

Fit Model 4

```
mod4 <- lm(data=sentence, lnyears ~ primlev0 + primlev02 + nsecond + seclev + seclev2 + nprior + priorlev + priorlev2 +
black + afro_m + babyface_m + attract_m)
ols_regress(mod4)
```

Model Summary

R	0.768	RMSE	0.700
R-Squared	0.589	Coef. Var	62.071
Adj. R-Squared	0.565	MSE	0.490
Pred R-Squared	0.527	MAE	0.526

RMSE: Root Mean Square Error
MSE: Mean Square Error
MAE: Mean Absolute Error

ANOVA

	Sum of Squares	DF	Mean Square	F	Sig.
Regression	142.910	12	11.909	24.282	0.0000
Residual	99.561	203	0.490		
Total	242.471	215			

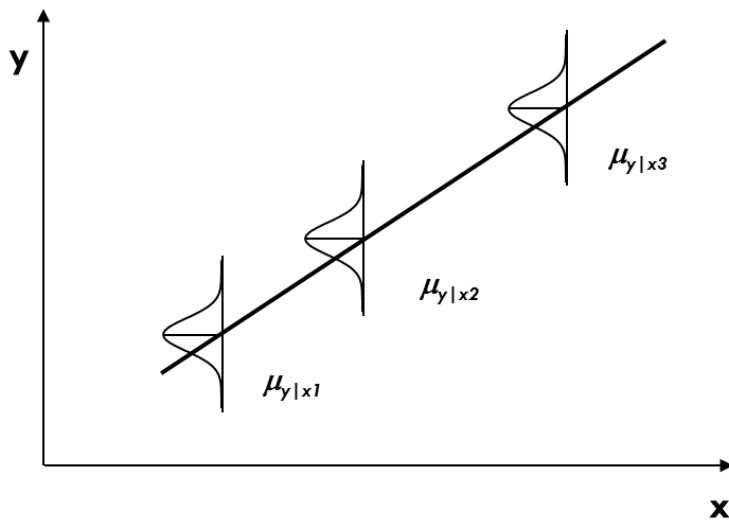
Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
(Intercept)	0.514	0.319		1.612	0.108	-0.115	1.143
primlev0	-0.138	0.115	-0.269	-1.199	0.232	-0.364	0.089
primlev02	0.039	0.010	0.868	3.776	0.000	0.018	0.059
nsecond	0.059	0.014	0.211	4.256	0.000	0.031	0.086
seclev	-0.111	0.061	-0.268	-1.811	0.072	-0.231	0.010
seclev2	0.022	0.008	0.411	2.677	0.008	0.006	0.038
nprior	0.022	0.036	0.039	0.604	0.547	-0.049	0.092
priorlev	-0.023	0.088	-0.050	-0.262	0.794	-0.197	0.151
priorlev2	0.004	0.012	0.055	0.321	0.748	-0.020	0.028
black	-0.327	0.142	-0.154	-2.304	0.022	-0.606	-0.047
afro_m	0.096	0.041	0.159	2.328	0.021	0.015	0.177
babyface_m	0.033	0.044	0.034	0.743	0.458	-0.055	0.121
attract_m	-0.017	0.055	-0.015	-0.309	0.758	-0.126	0.092

Afrocentric features continued to predict the log of sentence length when these variables were controlled. Therefore, babyface and attractiveness are not viable explanations. As a result, Model 3 is considered the final model.

Before accepting that Model 3 as a well-fitting model that meets the assumptions of a linear regression model, we need to take a closer look at the fitted model. In the rest of this unit, we will examine the fit of Model 3.

Assumptions of Fitting an OLS Regression Line



1. At each value of x , there is a distribution of y . These distributions have a mean $\mu_{y|x}$ and a variance of $\sigma^2_{y|x}$.

2. The relationship between x and y is linear. The means of each of these distributions, the $\mu_{y|x}$'s, may be joined by a straight line.

3. Homoscedasticity. The variances of each of these distributions, the $\sigma^2_{y|x}$'s, are equivalent.

4. Independence of observations. At each given value of x (at each x_i), the values of y (y_i 's) are independent of each other.

5. Normality – for inference only
At each given value of x (at each x_i), the values of y (the y_i 's) are normally distributed.

Note, $\sigma^2_{y|x}$ means “the variance of y given (or controlling for) x .”

Let's expand a bit on these basic tenants. Building on Assumption 2, we seek to achieve both linearity and additivity of the relationship between the dependent (y) and the independent variables (the set of x variables). Specifically, we assume that:

- The predicted values (\hat{y} -hats/fitted values) are a straight-line function of each predictor, holding all other predictors constant (the relationships are linear, not curvilinear).
- The slope representing each predictor does not depend on the values of the other variables (all relevant interactions are included in the model).


Tools for Evaluating Model Fit

Recall that when we fit an OLS regression model, each case receives a residual (e_i), which is equal to the case's observed y score minus \hat{y} (i.e., the predicted y given the case's score on each predictor in the model). We will use the residuals to check several assumptions. In addition, several other values, discussed below, will assist us in determining if we have outliers or influential cases later in the unit. We can use the `augment` function of the `modelr` package to obtain these helpful statistics after fitting a model.

Augment the dataset to pull in the fit and diagnostic information

```
fit_sentence <- augment(mod3, data=sentence)
```

These are statistics that are generated when we fit a linear model. The `augment` function pulls them from the `mod3` object and adds them to our dataframe.



id	years	lnyears	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd	.std.resid
1	3.83	1.34286480	0.8865945	0.11889733	0.456270343	0.02902035	0.6989014	1.195894e-03	0.663431949
2	11.50	2.44234704	2.2802697	0.13718201	0.162077296	0.03863249	0.6995572	2.049198e-04	0.236840882
3	4.33	1.46556754	1.6162629	0.12950184	-0.150695353	0.03442788	0.6995705	1.564971e-04	-0.219728659
4	17.17	2.84316367	2.5046910	0.18141077	0.338472639	0.06755921	0.6992224	1.661326e-03	0.502217845
5	14.17	2.65112705	0.7840683	0.13372577	1.867058758	0.03671036	0.6868589	2.573692e-02	2.725578818
6	4.42	1.48613970	1.7757483	0.22086190	-0.289608622	0.10013825	0.6993263	1.935696e-03	-0.437424191
7	99.00	4.59511985	3.2138957	0.21814266	1.381224168	0.09768764	0.6922065	4.271888e-02	2.083362876

`.fitted` is \hat{y} (i.e., the fitted or predicted value)

`.se.fit` is the standard error of the fitted value

`.resid` is the residual

`.hat` is the diagonal of the hat matrix (a matrix generated when we fit an MLR), it represents the case's leverage in the model

`.sigma` is an estimate of the residual standard deviation (RMSE) when the corresponding observation is dropped from model

`.cooksd` is Cook's D, a measure of the case's influence in the model

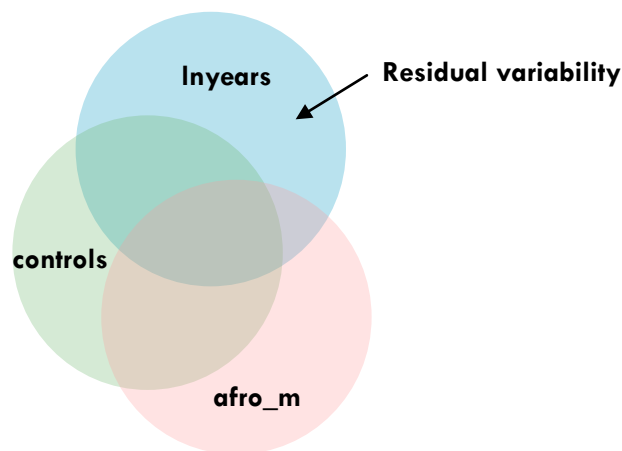
`.std.resid` is the case's standardized residual (a scaled version of the residual), i.e., a z-score of the residual

Linearity and Additivity: Assumption Checking & Remedies

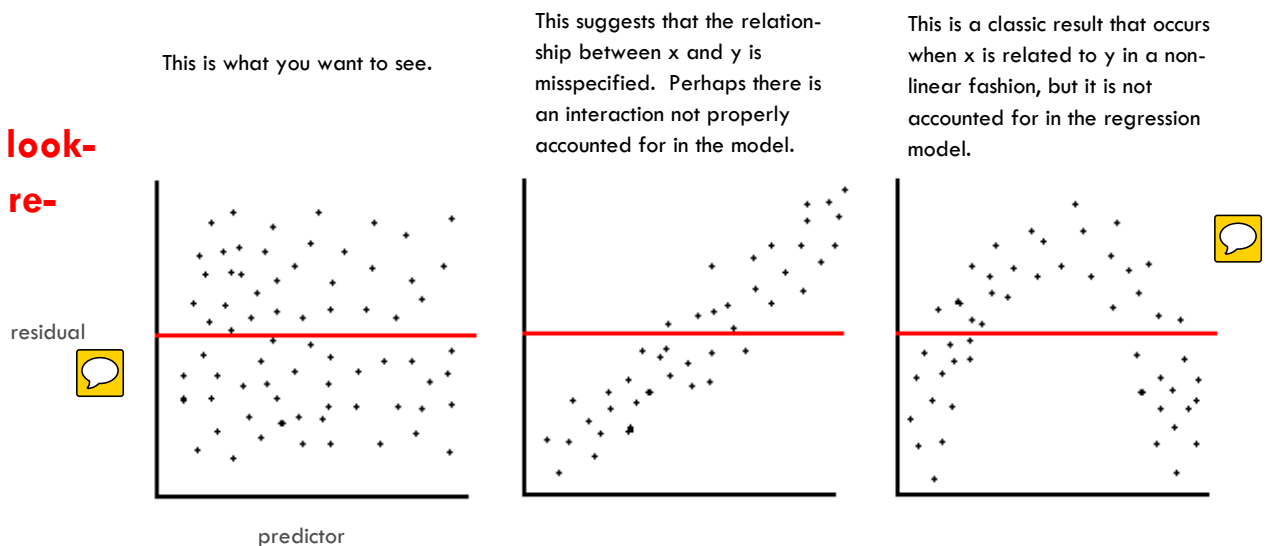
An OLS regression model assumes that the form of the relationship between each predictor variable (adjusting for all other predictors) and the outcome is linear. That is, that the model is linear in the parameters. To check this assumption, we can:

1. Plot the residuals against each predictor.
2. Plot the residuals against the predicted value (i.e., \hat{y} , fitted value).
3. Examine tests of curvature.
4. Create component + residual plots (also called partial-residual plots) for each predictor.

The first two techniques rely on residuals. The variability of the residuals represents the variance in the outcome (y) that is left over after we fit a linear regression model. Therefore the residuals should only represent variability in y that has nothing to do with any of the x variables. We can check to see if this is the case by plotting the residuals against each x , as well as the \hat{y} . If our model is correct, then we should **not see any systematic pattern** in these plots. Rather, each plot should look like random noise. If we do see a pattern, then this indicates that we may have a problem with our model.



What are we looking for in the residual plots?



Residuals Against Predictors and y-hat

Plot residuals against each predictor & y-hat

```
residualPlots(mod3, ask = FALSE, id.n = 3, labels.id = names(id), id.cex = 1.25, id.col = "blue", layout = c(1,1))
```

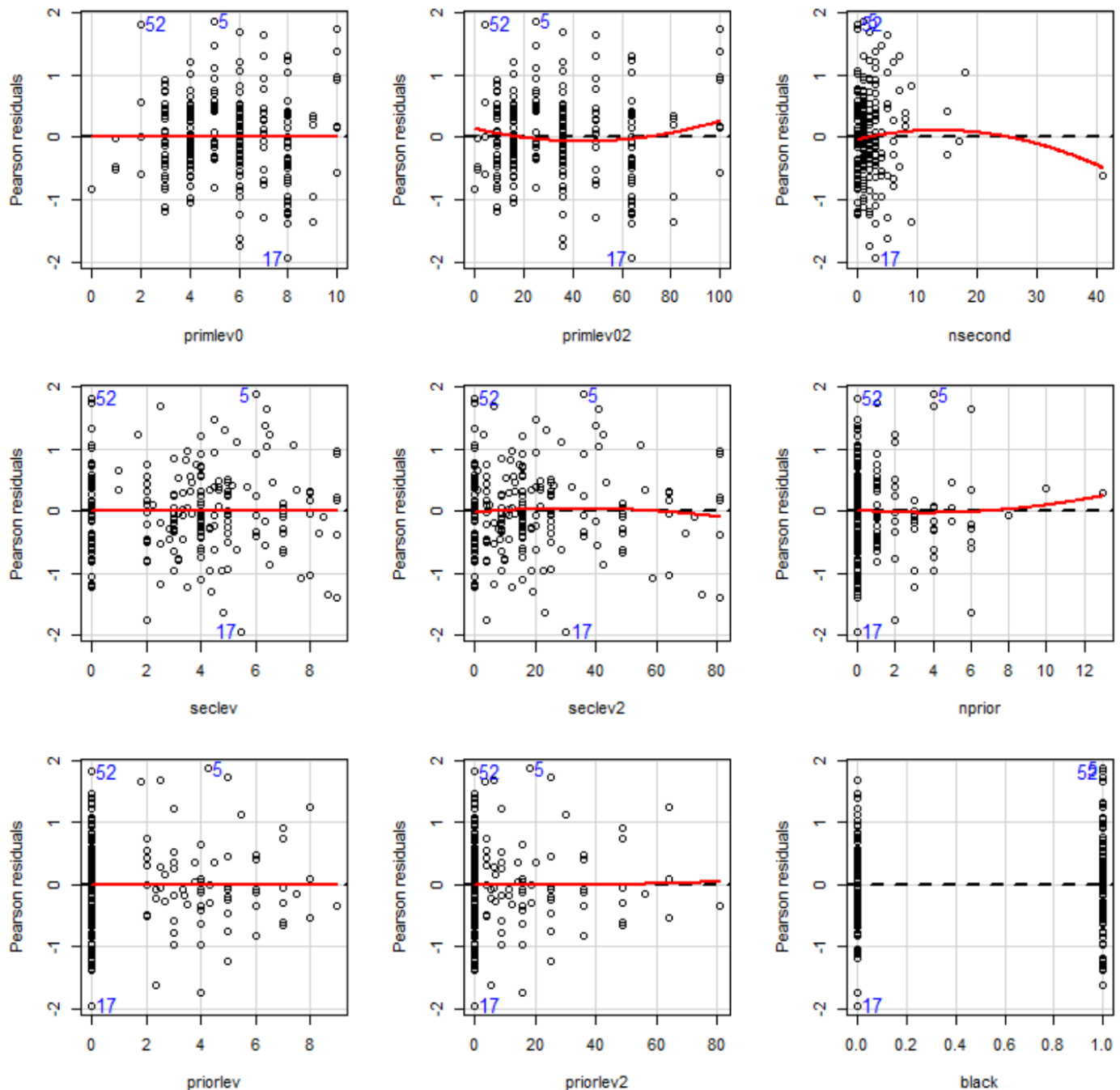
We will use the car (Companion for Applied Regression) package to examine most of our assumptions. Information about the package is here: <https://cran.r-project.org/web/packages/car/car.pdf>. The package's model checking functions are generally set up in the same way. First, provide the model name (mod3 in our case). Next, list ask = FALSE. This is needed because some of the car functions can be used interactively, but this feature hasn't been updated to work with notebooks yet. Therefore, ask = FALSE shuts off the interactivity.

The car package will label the most extreme points, and so the next four arguments listed (id.n, labels.id, id.cex, and id.col) provide information for how this is displayed (number of points to label, the variable that denotes the label, the size of the label, and the color of the label respectively).

The residualPlots function will plot the residual against each predictor and y-hat, so the final argument provides the desired layout of the graphs. Here, I provide c(1,1) for 1 row, 1 column. If you instead provided c(3,1), you'd get 3 rows, 1 column. Note that you could request the plot for just one of the predictors, in this case use the term argument. For example, term = "afro_m" would provide just the single plot for the residual against afrocentric features.

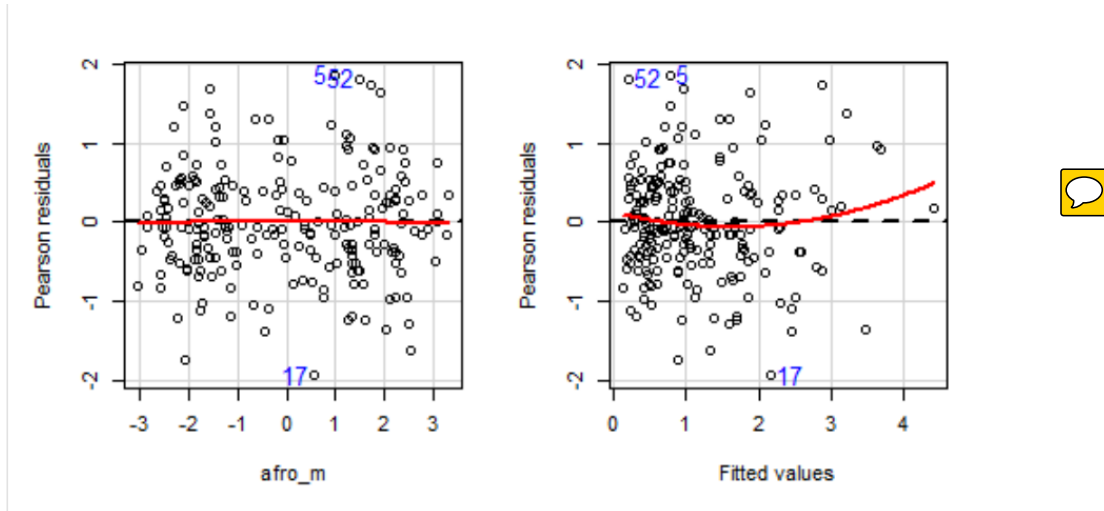
Residuals Against Control Variables

The plots generated by `residualPlots` are scatterplots with the standardized residual on the y-axis (the package calls these Pearson residuals) and the predictor on the x-axis. We hope to see no pattern. Since we requested `id.n = 3`, the three most extreme points (in terms of the absolute size of the residual) are plotted on each graph. The red line in these plots is a quadratic trend line—to the extent that it picks up a curvilinear relationship, then there may be a non-linear effect not being accounted for in the model. These look like reasonable plots—`primlev02`, `nsecond`, and `nprior` seem to have a bit of a curve, and `nsecond` seems to be driven by an outlier (an extreme value on `nsecond`).



Residuals Against Key Predictor (afro_m) and y-hat

Here are the same scatterplots for afro_m (our key predictor) and the y-hat (i.e, the predicted or fitted values) from our regression model. For both, we hope to see no discernable pattern. The residuals against afro_m look solid — there is no discernable pattern and no curvilinear relationship. For the fitted values, there doesn't seem to be a pattern, but there is a bit of an uptick to the quadratic line toward larger values of y-hat.



A Significance Test for Presence of Curvature

The residualPlots function also gives a significance test to indicate whether any of the red overlay lines are picking up a statistically significant degree of curvature. We get one for each predictor, and the last one (labeled Tukey test) corresponds to the red line on the y-hat (fitted values) plot. If the p-value is less than alpha, then a significant degree of curvature is detected. This is the case for primlev02. There may be a nonlinear relationship between primlev02 and lnyears (adjusting for the other variables) that we need to account for in the model. The significance test for the Tukey test is borderline, which is in correspondence with the look of the fitted values plot.

	Test stat	Pr(> t)
primlev0	0.385	0.700
primlev02	3.016	0.003
nsecond	-1.138	0.256
seclev	-1.017	0.310
seclev2	-1.266	0.207
nprior	0.663	0.508
priorlev	-0.711	0.478
priorlev2	0.222	0.824
black	0.413	0.680
afro_m	-0.087	0.931
Tukey test	1.819	0.069

Component + Residual Plot

While assessment of the residual plots against each predictor are helpful, they don't fully capture the multivariate nature of the MLR model. Component + residual plots can help us in this case. Here, the partial residual, rather than the usual residual, is plotted on the y-axis. To obtain the partial residual for x_1 , you multiply the value of the predictor of interest times the regression slope for the predictor and add this quantity to the residual; that is, $b_{x_1}x_{1i} + \text{residual}_i$. R creates this quantity for you. Component + residual plots can be very useful to determine if a transformation is needed. Because we've added back the systematic component of the relationship, we expect to see a linear relationship between each predictor and the corresponding partial residual. If there is a curvilinear relationship, we can match it to the Rule of the Bulge diagram to determine what type of transformation is needed.

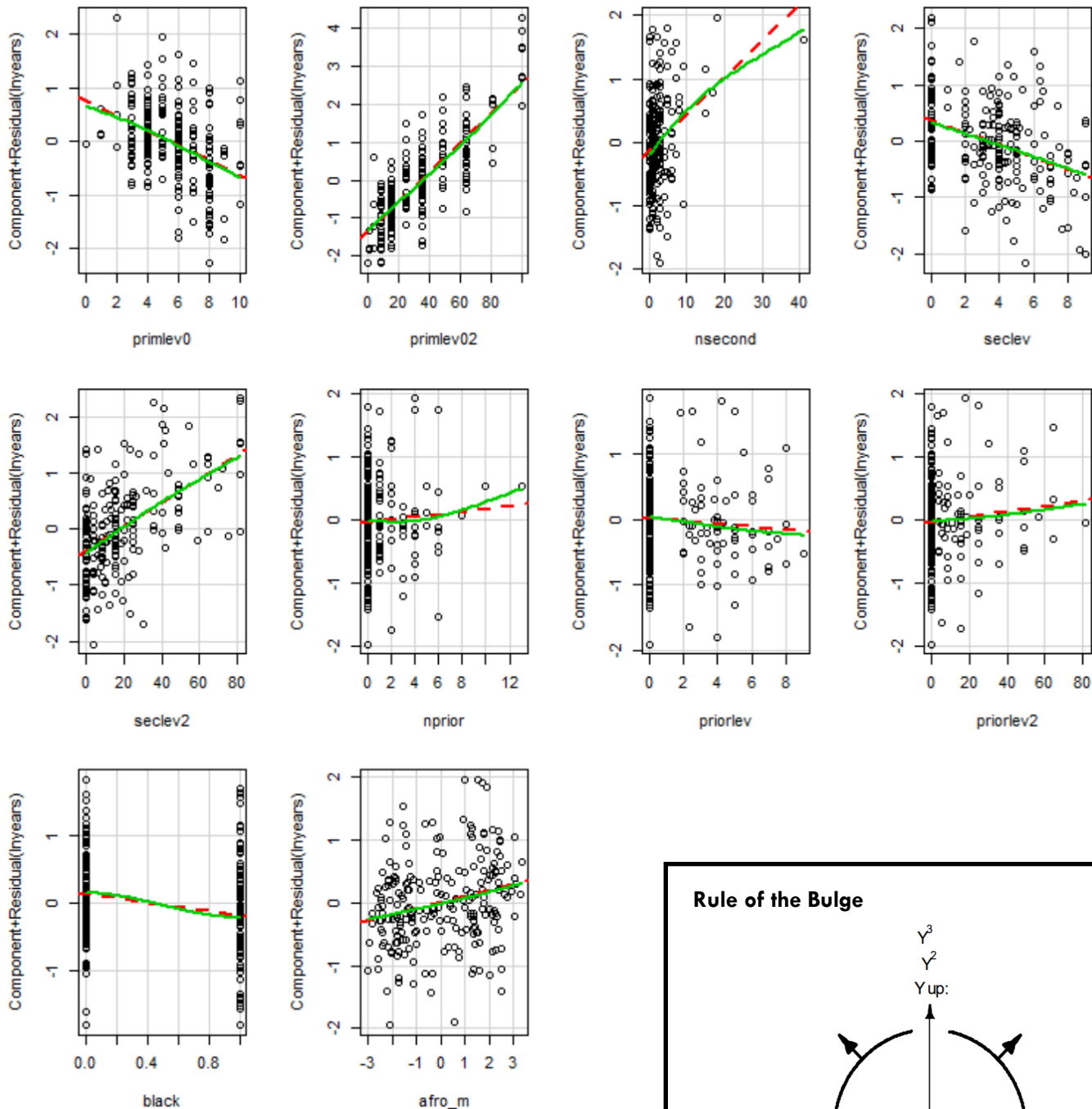
In the component + residual plots created by the car package, each plot will include a dashed red line and a green line. The dashed red line is the linear solution to the partial relationship, the green line is a loess smooth. A loess smooth identifies a smooth curve by following the empirical concentration of the points in the plot. The smoothed "line" passes through the most dense areas of the data. To the extent that the loess smooth suggests a curvilinear relationship is a better fit, then a transformation may be needed.

Please note that the loess smooth uses a "span" to follow the curve—the default is 2/3rds of the value of x . If there are sparse places along the x-axis, then you may need to increase the span (see the code below). If there are large sparse areas, you may not be able to obtain a good loess curve (see the error message for `nsecond` when you execute the code).

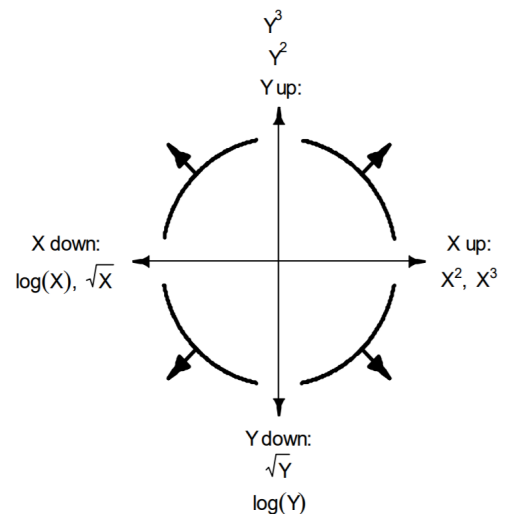
Component plus residual plots

```
crPlots(mod3, ask = FALSE, terms = "primlev0")
crPlots(mod3, ask = FALSE, span = 1, terms = "primlev02")
crPlots(mod3, ask = FALSE, terms = "nsecond")
crPlots(mod3, ask = FALSE, terms = "seclev")
crPlots(mod3, ask = FALSE, span = 1, terms = "seclev2")
crPlots(mod3, ask = FALSE, span = 1, terms = "nprior")
crPlots(mod3, ask = FALSE, span = 1, terms = "priorlev")
crPlots(mod3, ask = FALSE, span = 1, terms = "priorlev2")
crPlots(mod3, ask = FALSE, terms = "afro_m")
```


Component + Residual Plots



Rule of the Bulge



You can see that *nsecond* and *nprior* seem to depart from linearity. The curve suggested by the green line for *nsecond* matches the upper left quadrant of the Rule of the Bulge. If we favor transforming *x* (recall that *y* was already transformed), then going down in *x* (e.g., $\log(x)$) might be helpful. The green line for *nprior* matches the lower right quadrant of the Rule of the Bulge, so going up in *nprior* might be helpful. Note that this technique isn't useful for binary predictors (e.g., *black*).

Remedy — Violation of the Assumptions of Linearity & Additivity

If you find that the linearity assumption is violated, then you have a non-linear relationship that needs to be correctly specified. This was the topic of Unit 8, so you have the tools, knowledge, and skills to do this. Apply the techniques presented in Unit 8 and then re-evaluate the assumptions to ensure the problem is solved.

If you find that the assumption of additivity is violated, then you can examine whether the inclusion of interactions remedies the situation. Add the interaction term(s) to the model and then re-evaluate the assumptions to ensure the problem is solved.

Homoscedasticity: Assumption Checking & Remedies

An OLS regression model assumes that the variance of the errors is constant across the linear combination of the predictors. If the variance of the residuals is related to any of the predictors or to the predicted values (i.e., \hat{y} -hats, the fitted values), then this assumption may be violated. Violation of this assumption compromises the accuracy of our significance tests because the standard errors might be incorrect. To check this assumption, we can:

1. Plot the residuals and predicted value against each predictor (i.e., the same residualPlot that we created to examine linearity).
2. Examine the score test for non-constant error variance.

Test of non-constant error variance

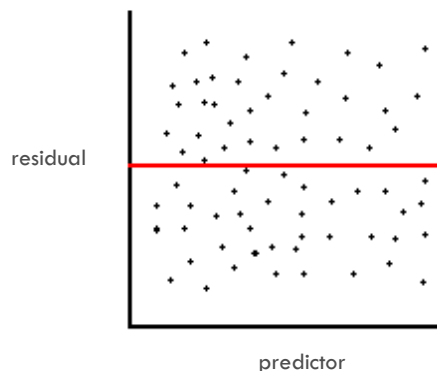
```
ncvTest(mod3)
```

If this test is significant (i.e., p-value is less than alpha (e.g., .05)), then the model likely violates the assumptions of homogeneity of variance.

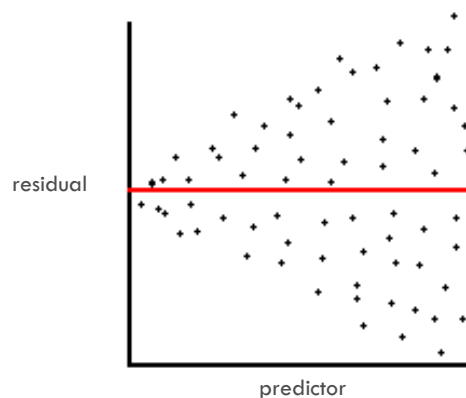
What are we looking for in the residual plots in terms of homogeneity?

We want to see that when the residual is plotted against each predictor and the fitted value, that the variability of the residuals is similar across the range of the value on the x-axis.

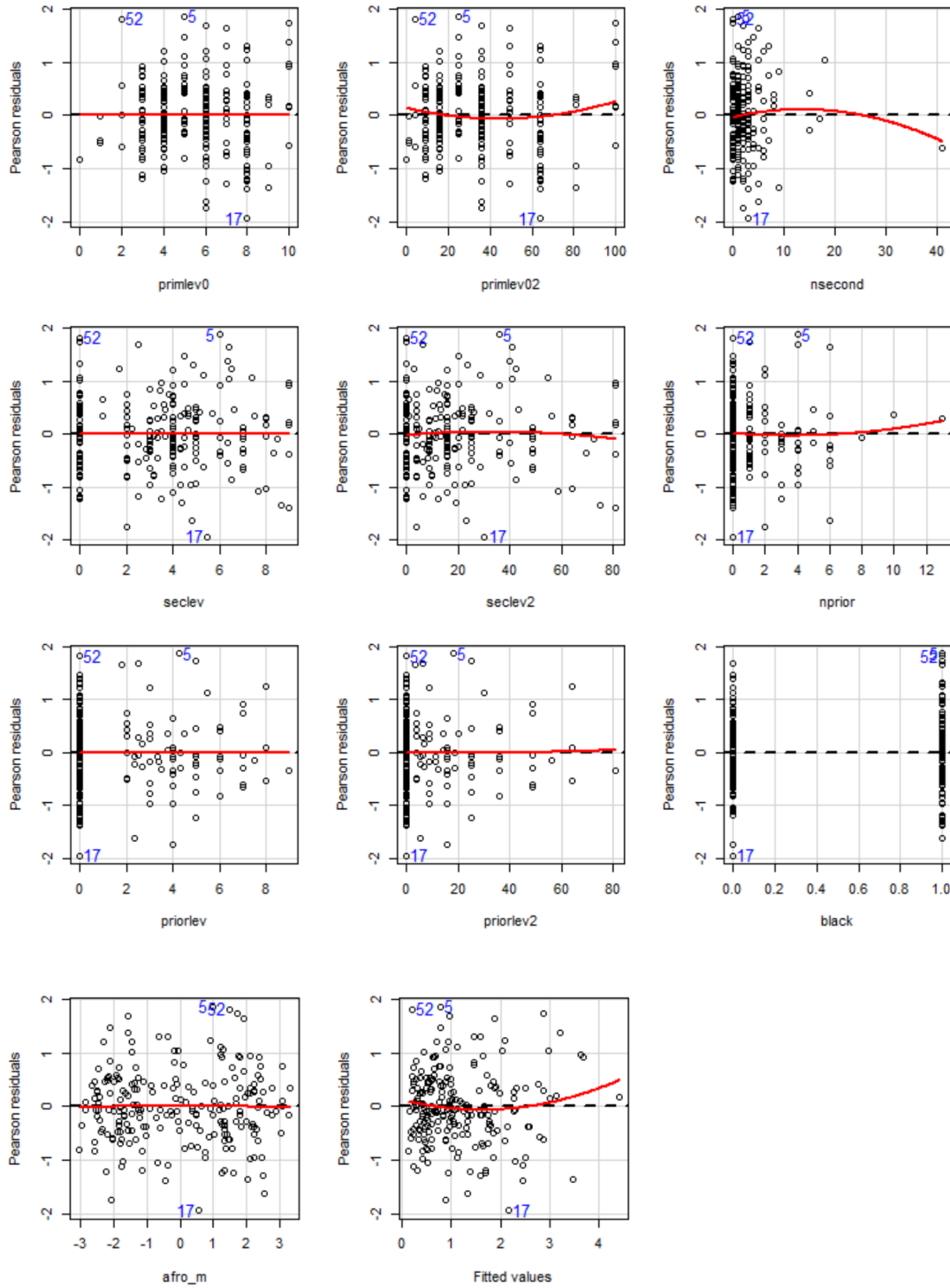
This is what you want to see.



This is indicative of a problem. As we move across the x-axis, the variability of the residuals increases. This means that we do a worse job of predicting the outcome as x gets larger.



Residual Plots to Examine Homogeneity of Variance



The residual plot against the predictors and the fitted values seems okay for most plots, but not good for nsecond and nprior, and maybe priorlev2.

The non-constant variance score test is significant, suggesting that the assumption of homogeneity of variance is violated.

Non-constant Variance Score Test
 Variance formula: ~ fitted.values
 Chisquare = 7.491156 Df = 1 p = 0.006200274

Remedy—Violation of Homoscedasticity Assumption

If you find that the homogeneity of variance assumption is violated, you can consider one or more of the following:

1. Consider missing covariates. Is there a predictor that you can include that would do a better job of predicting the outcome in the regions with large residuals?
2. Consider transformations and/or missing interactions (i.e., perhaps your model isn't properly specified).
3. Use robust standard errors.

Robust Standard Errors in R

In the presence of heteroscedastic errors (i.e., violation of the assumption of homogeneity), the OLS regression coefficient estimates are unbiased, but the standard errors can be biased. Of course, since we divide the estimate by the standard error to obtain t^* and the p -value (i.e., to use for hypothesis testing), then biased standard errors are undesirable. A common solution to this problem is to replace the usual standard errors with standard errors that are robust even when the assumption of homogeneity of error variances is violated. We can accomplish this in R with the code below. The `hccm` (heteroskedasticity consistent covariance matrix) is from the `car` package, and uses the White (1980) method of correcting the heteroskedastic errors. The `coeftest` function is from the `lmtest` package and it pulls in the initial model, replaces the usual standard errors with `hccm` standard errors, and outputs the new results.

Use a heteroskedasticity consistent covariance matrix to compute standard errors

```
mod3_hccm <- hccm(mod3)
coeftest(mod3, vcov=mod3_hccm)
```

t test of coefficients:

```

      Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.5062339  0.3816637  1.3264 0.186187
primlev0     -0.1382505  0.1372880 -1.0070 0.315117
primlev02    0.0389767  0.0127334  3.0610 0.002501 **
nsecond      0.0580221  0.0211217  2.7470 0.006549 **
secllev      -0.1100275  0.0660843 -1.6650 0.097449 .
secllev2      0.0214930  0.0089746  2.3949 0.017527 *
nprior       0.0209189  0.0329673  0.6345 0.526439
priorlev     -0.0236249  0.0908310 -0.2601 0.795050
priorlev2    0.0039953  0.0125237  0.3190 0.750035
black        -0.3218678  0.1543768 -2.0849 0.038312 *
afro_m       0.0922101  0.0414357  2.2254 0.027146 *
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



We estimate and then use the standard errors from the heteroscedasticity-corrected covariance matrix. The results here present the new standard errors (note that the coefficient estimates remain the same). The standard errors have changed a bit, but the overall conclusions of the model are the same. This indicates that the violation of this assumption didn't have a large impact on the OLS model.

Model 3 results
with usual stand-
ard errors.

Parameter Estimates							
model	Beta	Std. Error	Std. Beta	t	sig	lower	upper
(Intercept)	0.506	0.316		1.602	0.111	-0.117	1.129
primlev0	-0.138	0.114	-0.270	-1.211	0.227	-0.363	0.087
primlev02	0.039	0.010	0.876	3.831	0.000	0.019	0.059
nsecond	0.058	0.014	0.208	4.233	0.000	0.031	0.085
secllev	-0.110	0.061	-0.267	-1.812	0.071	-0.230	0.010
secllev2	0.021	0.008	0.409	2.676	0.008	0.006	0.037
nprior	0.021	0.036	0.037	0.588	0.557	-0.049	0.091
priorlev	-0.024	0.088	-0.051	-0.269	0.788	-0.197	0.150
priorlev2	0.004	0.012	0.055	0.325	0.746	-0.020	0.028
black	-0.322	0.141	-0.151	-2.280	0.024	-0.600	-0.044
afro_m	0.092	0.040	0.153	2.306	0.022	0.013	0.171

Normality of Errors: Assumption Checking & Remedies

An OLS regression model assumes that the errors are normally distributed. Violation of this assumption compromises the accuracy of our significance tests.

To check this assumption, we can:

1. Make a histogram of the residuals.
2. Make a QQ plot of residuals (plot empirical quantiles of the studentized* residuals, against the quantiles of a normal distribution).
3. Conduct a formal test of the normality of the residuals.

Density plot of residuals

```
ggplot(data = fit_sentence, aes(x = .std.resid)) +  
  geom_density() +  
  stat_function(fun = dnorm,  
               lwd = 2,  
               col = "red") +  
  labs (title = "Density plot of standardized residuals against standard normal distribution (red)")
```

QQ plot of residuals

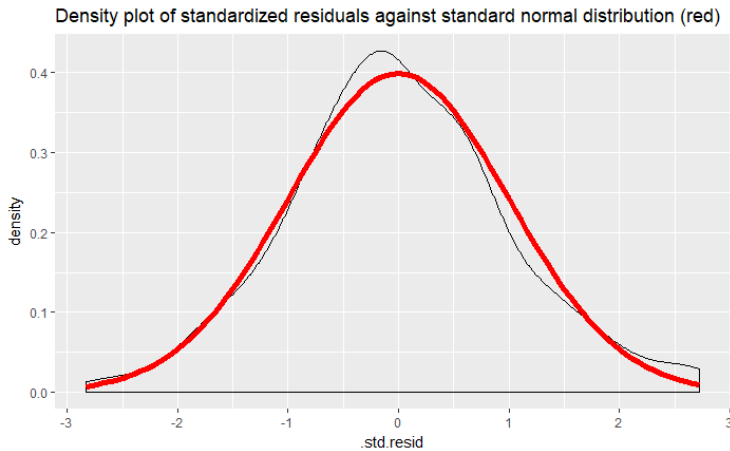
```
qqPlot(mod3, ask = FALSE, id.n = 3, labels.id = names(id), id.cex= 1.25, id.col = "blue")
```

Shapiro-Wilk test for normality

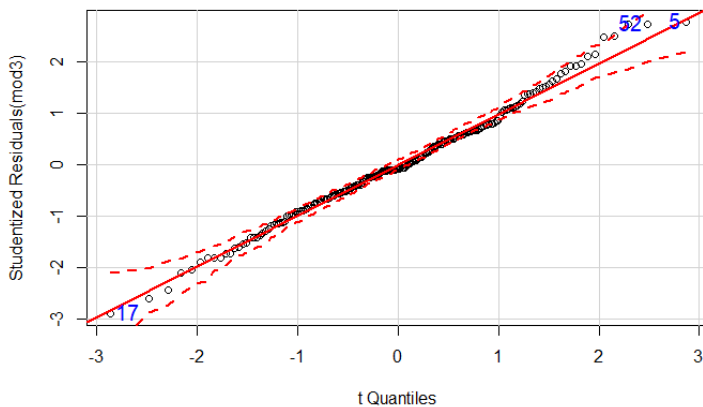
```
shapiro.test(fit_sentence$.resid)
```

*The car package uses the studentized residuals for the QQ plot. The studentized residuals take into account that the raw residuals may have substantially different variances. For example, the variability of cases close to the sample mean for the predictors have a smaller variance compared to cases further away from the mean. The studentized residual takes this change in variability into account by dividing the raw residual by an estimate of the standard deviation of the residual at that point. They are typically similar to the standardized (Pearson) residuals.

Assessment of Normal Residuals



With the density plot, we want to see that the residuals from the model follow a normal distribution (the red distribution). Remember that with a normal distribution, we expect a bell-shaped symmetric curve, with about 95% of the residuals falling within plus/minus 2 standard deviations of the mean. This plot looks reasonable.



The QQ plot is another way of looking at the normality of the residuals. The red solid line represents what we'd expect if the model residuals follow a perfect normal distribution. To the extent that the data points (black dots) deviate from this line (and in particular that they are outside of the red dotted confidence band), then our residuals are not normal. Again, this plot looks pretty good.

shapiro-wilk normality test

```
data: fit_sentence$.resid
W = 0.99318, p-value = 0.4224
```

This is a formal test of the normality of the residuals. If the p-value is less than alpha (e.g., .05), then there is an indication that the residuals are not normal, and therefore violate the assumption of normally-distributed residuals. The Shapiro-Wilk test is not significant, so it looks like this model is in good shape on this front.

Remedy—Violation of Normally Distributed Errors

If you find that the residuals are not normally distributed, you can consider one or more of the following:

1. Consider transformations and or missing interactions (i.e., perhaps your model isn't properly specified). Often times what fixes up the other violations, also fixes other violations.
2. Bootstrap the estimates and standard errors.

An Introduction to Bootstrapping

Bootstrapping is an alternative method for hypothesis testing. The method builds a sampling distribution for an estimate of interest. By using this empirical distribution rather than the sampling distribution, one does not need to rely on the assumption of normality as bootstrapping does not require this assumption.

A bootstrapped distribution is formulated by resampling from the observed data. The method first draws a sample of size n (with replacement) from the observed data. This procedure is repeated many, many (i.e., , 5000 or more) times. Once the bootstrap resamples are created, the estimates of interest are computed in each of the bootstrap samples. From these estimates, the estimates of interest and associated standard errors are calculated.

Bootstrapping the Regression Model in R

Let's take a look at how we could bootstrap Model 3.

```
bs <- function(formula, data, indices){
  d <- data[indices,]
  regmodel <- lm(formula, data = d)
  return(coef(regmodel))
}

results <- boot(data=sentence, statistic = bs,
  R = 5000, formula = lyears ~ primlev0 + primlev02 +
nsecond + seclev + seclev2 + nprior +
priorlev + priorlev2 + black + afro_m)

# get bootstrap confidence intervals for each estimate
boot.ci(results, type="bca", index=1) # intercept
boot.ci(results, type="bca", index=2) # primlev0
boot.ci(results, type="bca", index=3) # primlev02
boot.ci(results, type="bca", index=4) # nsecond
boot.ci(results, type="bca", index=5) # seclev
boot.ci(results, type="bca", index=6) # seclev2
boot.ci(results, type="bca", index=7) # nprior
boot.ci(results, type="bca", index=8) # priorlev
boot.ci(results, type="bca", index=9) # priorlev2
boot.ci(results, type="bca", index=10) # black
boot.ci(results, type="bca", index=11) # afro_m
```

← Don't change anything here, this just creates a function.

← Change the underlined parts for a new example.

← These are listed in order of their entry on the formula line.

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 1)

Intervals :
Level Bca
95% (-0.1003, 1.3538)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 2)

Intervals :
Level Bca
95% (-0.4425, 0.0888)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 3)

Intervals :
Level Bca
95% (0.0176, 0.0659)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 4)

Intervals :
Level Bca
95% (0.0340, 0.1005)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 5)

Intervals :
Level Bca
95% (-0.2311, 0.0179)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 6)

Intervals :
Level Bca
95% (0.0041, 0.0378)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 7)

Intervals :
Level Bca
95% (-0.0578, 0.0838)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 8)

Intervals :
Level Bca
95% (-0.2013, 0.1421)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 9)

Intervals :
Level Bca
95% (-0.0184, 0.0302)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 10)

Intervals :
Level Bca
95% (-0.6068, -0.0264)
Calculations and Intervals on Original Scale
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 5000 bootstrap replicates

CALL :
boot.ci(boot.out = results, type = "bca", index = 11)

Intervals :
Level Bca
95% (0.0126, 0.1723)
Calculations and Intervals on Original Scale

		Parameter Estimates							
		model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
Model 3 results with usual confi- dence intervals.	(Intercept)		0.506	0.316		1.602	0.111	-0.117	1.129
	primlev0		-0.138	0.114	-0.270	-1.211	0.227	-0.363	0.087
	primlev02		0.039	0.010	0.876	3.831	0.000	0.019	0.059
	nsecond		0.058	0.014	0.208	4.233	0.000	0.031	0.085
	seclev		-0.110	0.061	-0.267	-1.812	0.071	-0.230	0.010
	seclev2		0.021	0.008	0.409	2.676	0.008	0.006	0.037
	nprior		0.021	0.036	0.037	0.588	0.557	-0.049	0.091
	priorlev		-0.024	0.088	-0.051	-0.269	0.788	-0.197	0.150
	priorlev2		0.004	0.012	0.055	0.325	0.746	-0.020	0.028
	black		-0.322	0.141	-0.151	-2.280	0.024	-0.600	-0.044
	afro_m		0.092	0.040	0.153	2.306	0.022	0.013	0.171

The 95% CI for the slopes differ from the initial model, but the overall conclusions drawn from the model remain the same.

Independence of Residuals: Assumption Checking & Remedies

An OLS regression model assumes that the residuals are independent of one another. The most common sources of non-independence come from temporal dependency, repeated measures, or clustering of cases in groups.

Temporal dependency occurs when there is systematic change over time in the nature or characteristics of the cases (e.g., people in your study). For example, we might suspect that people in the PSY100 pool who wait until the end of the semester to earn their research credits are different than those who participate at the beginning. The people who finish early may tend to be more conscientious. If we are aware of clustering, we can control for it — for example, we can add date of participation as a predictor. And, since we adjust for the non-independence, the residuals become independent.

Repeated measures data (i.e., multiple assessments of the same person over time) or clustering of individuals in some upper level unit (e.g., companies, families, schools), requires multilevel modeling. If there are just a few clusters, you can create $k-1$ dummy codes (where k is the number of clusters), and add these as control variables in the model. There's a phenomenal suite of packages in R for multilevel modeling and we will explore these in our Introduction to Multilevel Modeling Unit later in the semester.

Remedy—Violation of Independent Errors

If you find that the residuals are not independent, you can consider one of the following methods:

1. Control for any dependencies (i.e., time, clusters, etc.)
2. Use multilevel models to analyze the data.

Unusual Cases: Detecting Them & Remedies

Unusual cases include outliers (i.e., cases with an unusual y -score based on their scores on the predictors), high-leverage cases (i.e., unusual based on their multivariate distribution of predictor scores—a strange combination of scores on the predictors), and influential cases (i.e., cases that have an exceptionally large impact on the regression estimates). Unusual cases can have a substantial impact on your model. In SLR, unusual cases are easy to detect with plots. But with MLR, particularly with many variables, it becomes quite difficult to detect them visually. There are a series of statistics that we can use to find unusual cases.

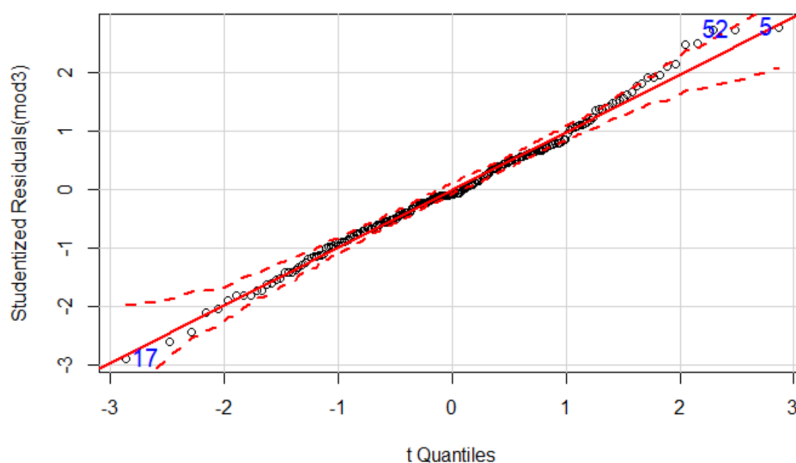
Let's start with OUTLIERS

Outliers are cases that don't follow the norm of the data. One way to detect an outlier is to examine the residuals. We can look at the QQ plot that we examined earlier to find potential outliers. Let's label the 3 most extreme points (i.e., with the largest studentized residuals). We can also conduct an outlier test. This test calculates the Bonferroni-corrected p -value (corrected for the many tests made when testing the size of each studentized residual) to test whether any residual is unexpectedly large.

Find cases with a large residual

```
# qqplot —identify cases with large studentized residuals
qqPlot(mod3, id.n=3, main="Distribution of Studentized Residuals")

# outlier test
outlierTest(mod3)
```



We expect that only about 5% of the standardized (or studentized residuals) will be outside the range of ± 2 , and 1% outside of ± 3 standard deviations from the mean (0 for residuals). This model seems in line with those expectations.

```
No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
      rstudent unadjusted p-value Bonferonni p
17 -2.882378      0.0043695      0.94381
```

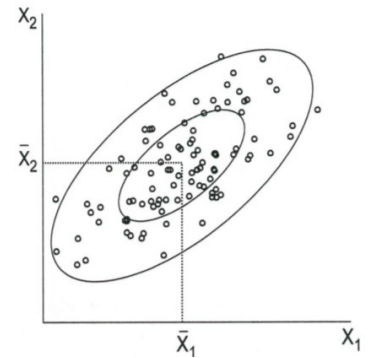


The p -value is not significant, indicating that this large residual is not unexpected.

Now, Consider HIGH LEVERAGE CASES

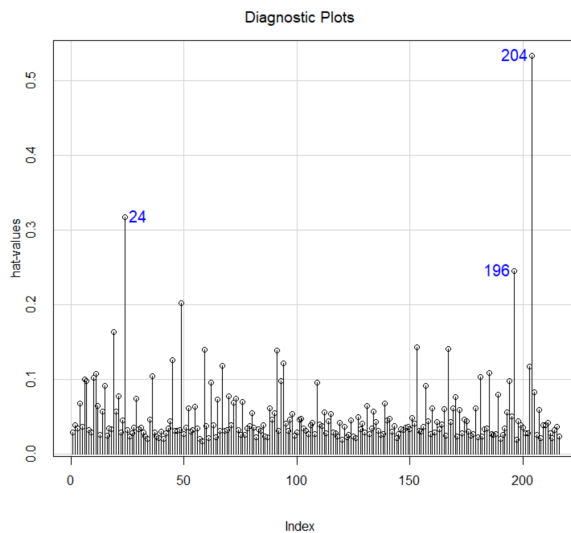
Leverage captures a case's extremity on the predictors in the model, taking into account the correlational pattern of the predictors. The leverage (also called hat value) measures the multivariate distance between the x values for a certain case and the multivariate center of all x variables (i.e., the centroid). Leverage values range from $1/n$ to 1. The larger the leverage, the greater the weight that case receives in calculating the predicted values of y (i.e., \hat{y} , fitted value). The average leverage is $(k+1)/n$, where k is the number of predictors and n is the sample size.

We can use an index plot for assessing leverage. An index plot lists each case (e.g., person) on the x -axis and the score of interest (e.g., leverage) on the y -axis. When evaluating the plot, take a look at where a gap occurs in the distribution of the leverage scores and examine the cases in the most extreme group.



Index plot for high leverage

```
inflIndexPlot(mod3, ask = FALSE, vars=c("hat"), id.n = 3, labels.id = names(id), id.cex= 1.25, id.col = "blue")
```

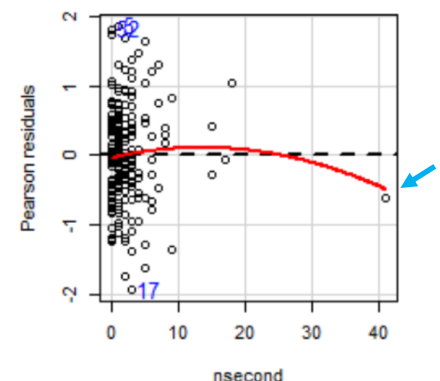


Three cases have relatively high leverage scores. One is particularly high — case 204.

id	years	black	afro	primlev	seclev	nsecond	anysec	priorlev	nprior	anyprior	attract	babyface	black.f
204	9.42	0	3.20588	5	5.330	41	1	0.00	0	0	2.94118	3.32353	White

This person has an extremely high score for nsecond (secondary offenses). In fact they have the highest score in the entire sample.

	vars	n	mean	sd	min	max
id	1	216	108.50	62.50	1.00	216.00
years	2	216	6.84	15.54	0.42	99.00
black	3	216	0.46	0.50	0.00	1.00
afro	4	216	4.53	1.77	1.49	7.86
primlev	5	216	6.55	2.07	1.00	11.00
seclev	6	216	3.40	2.57	0.00	9.00
nsecond	7	216	2.41	3.82	0.00	41.00
anysec	8	216	0.75	0.44	0.00	1.00
priorlev	9	216	1.43	2.29	0.00	9.00
nprior	10	216	0.95	1.90	0.00	13.00
anyprior	11	216	0.33	0.47	0.00	1.00
attract	12	216	3.21	0.91	1.43	6.51
babyface	13	216	4.04	1.10	1.66	7.09
black.f*	14	216	1.46	0.50	1.00	2.00



Last, Consider INFLUENTIAL CASES

A case that is both an outlier and has high leverage is influential (i.e., the case exerts influence on the regression estimates). This means that if we remove an influential case, the regression estimates change substantially. We'll consider Cook's Distance and DFBETAS.

Cook's Distance measures the influence that each case has on the estimation of \hat{y} (i.e., fitted or predicted value) for all cases in the sample. Cook's Distance can't be smaller than 0 and higher values are of more concern. Look for cases that stick out like a sore thumb in the index plot, and give them a closer look.

A DFBETAS value is calculated for each predictor and measures the extent to which each case influences the regression slope. DFBETAS is the standardized change in the slope when the case is removed from the dataset. Positive values indicate that omitting the case would increase the slope estimate, negative values indicate that omitting the case would decrease the slope estimate. Again, look for cases that stick out from the others in the index plot.

We can also look at the added variable plots. These plots can help us find jointly influential points. They plot the residual from a model where y is regressed on all predictors except the predictor of interest against the residual where the x of interest is regressed on all other predictors. If this sounds familiar, it's because we studied it last semester. Correlation of these residuals yields the partial correlation. You get one plot for each predictor. These plots show how the partial relationship is affected by certain cases.

We'll also check out a summary plot that combines leverage and influence, which can be extremely useful.

Index plot for Cook's Distance

```
inflIndexPlot(mod3, ask = FALSE, vars=c("Cook"), id.n = 2, labels.id = names(id), id.cex= 1.25, id.col = "blue")
```

Index plot for DFBETAS

```
dfbetasPlots(mod3, ask = FALSE, id.n = 3, labels.id = names(id), id.cex= 1.25, id.col = "blue", layout = c(4,3))
```

Added variable plots

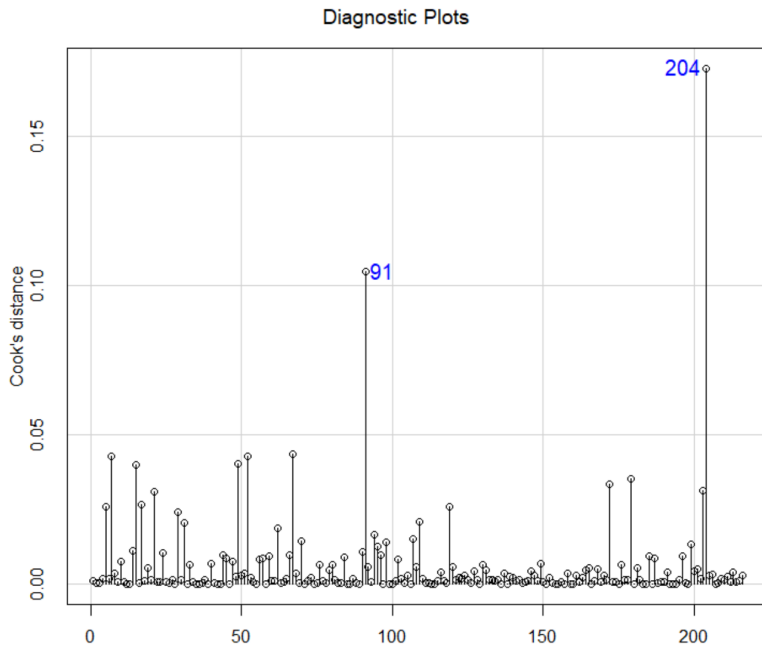
```
avPlots(mod3, ask = FALSE, id.n = 3, labels.id = names(id), id.cex= 1.25, id.col = "blue", layout = c(4,3))
```

Summary influence plot

```
influencePlot(mod3, ask = FALSE, labels.id = names(id), id.cex= 1.25, id.method = "noteworthy", id.col = "blue")
```

Each of the model checking functions label the points based on a default setting. You can learn about these settings here: <https://www.rdocumentation.org/packages/car/versions/2.1-6/topics/showLabels>.

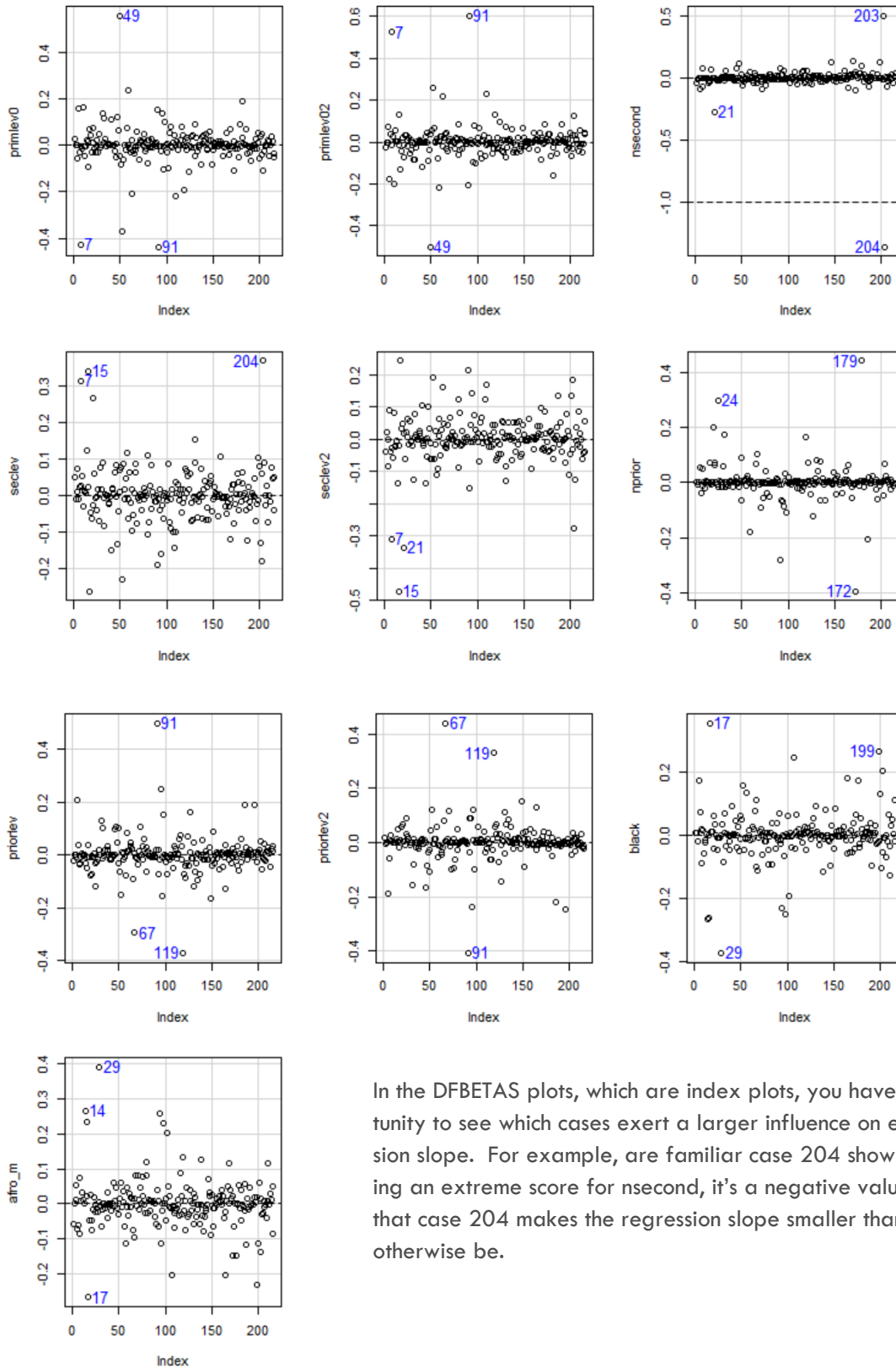
Cook's Distance



Two cases are much higher than the others on Cook's D. Case 204 is the highest, which is the same individual who was high on leverage. Now we know they have an unusual combination of scores on the predictor variables and/or extreme scores on predictor variable(s), and their presence in the model influences the predicted values of *lnyears* for all other cases.

DFBETAS

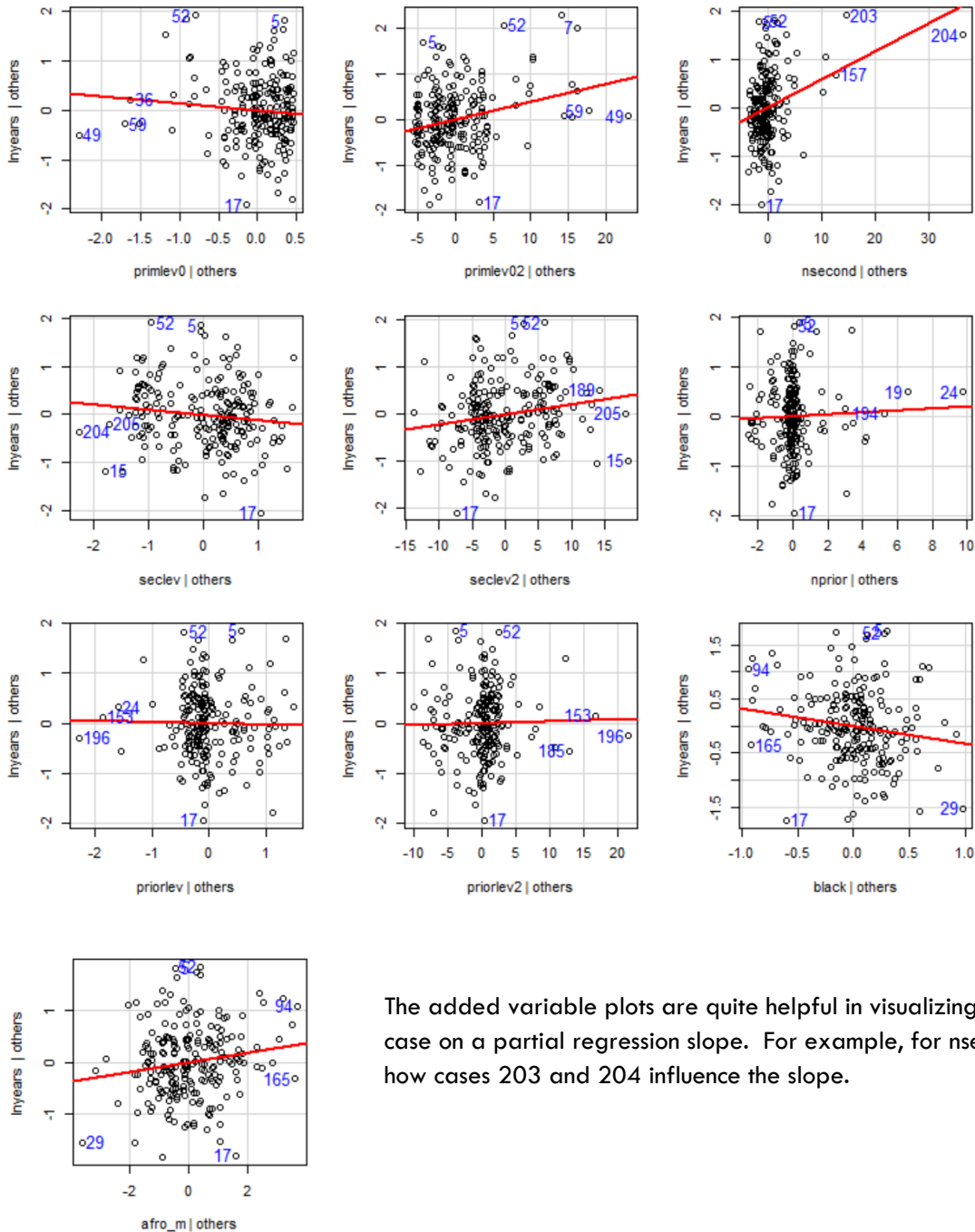
dfbetas Plots



In the DFBETAS plots, which are index plots, you have an opportunity to see which cases exert a larger influence on each regression slope. For example, are familiar case 204 shows up as having an extreme score for nsecond, it's a negative value indicating that case 204 makes the regression slope smaller than it would otherwise be.

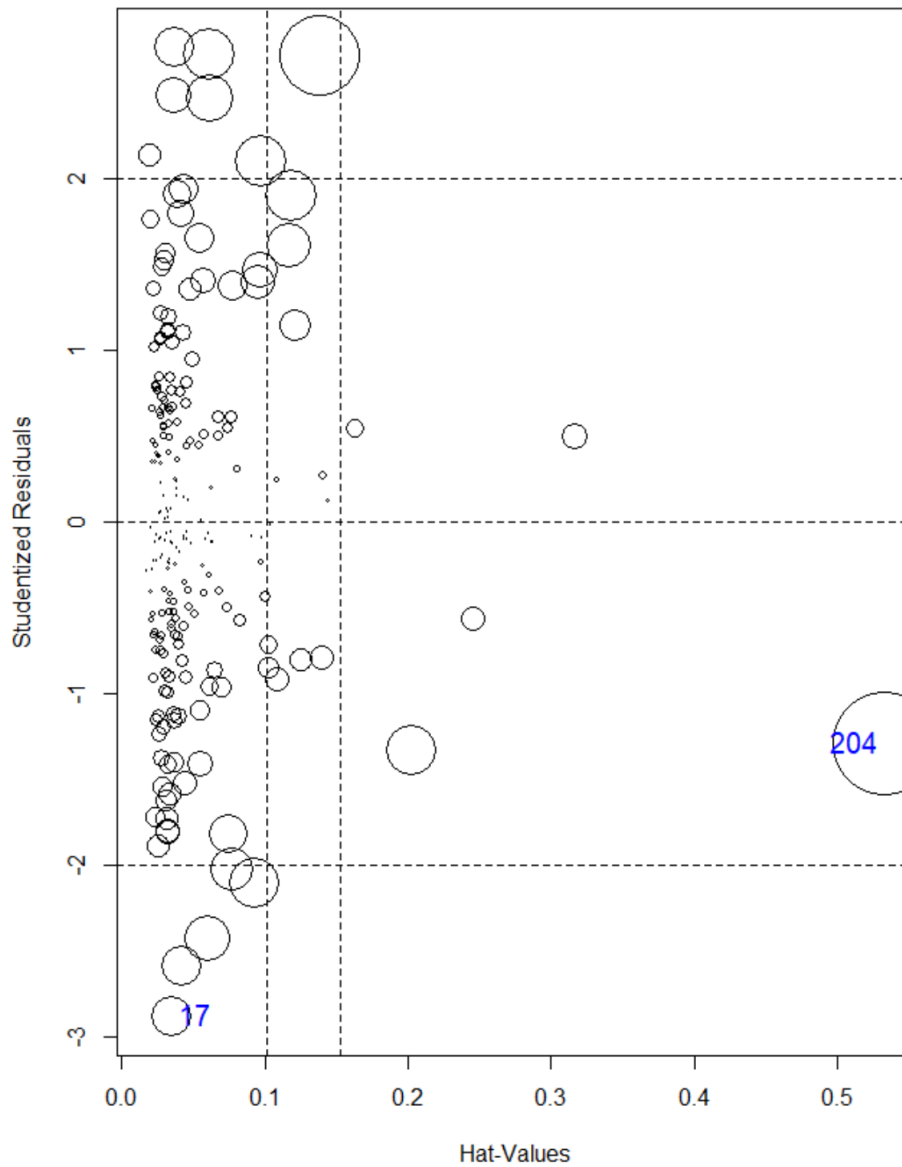
Added Variable Plots

Added-Variable Plots



The added variable plots are quite helpful in visualizing the potential impact of a case on a partial regression slope. For example, for *nsecond*, you can really see how cases 203 and 204 influence the slope.

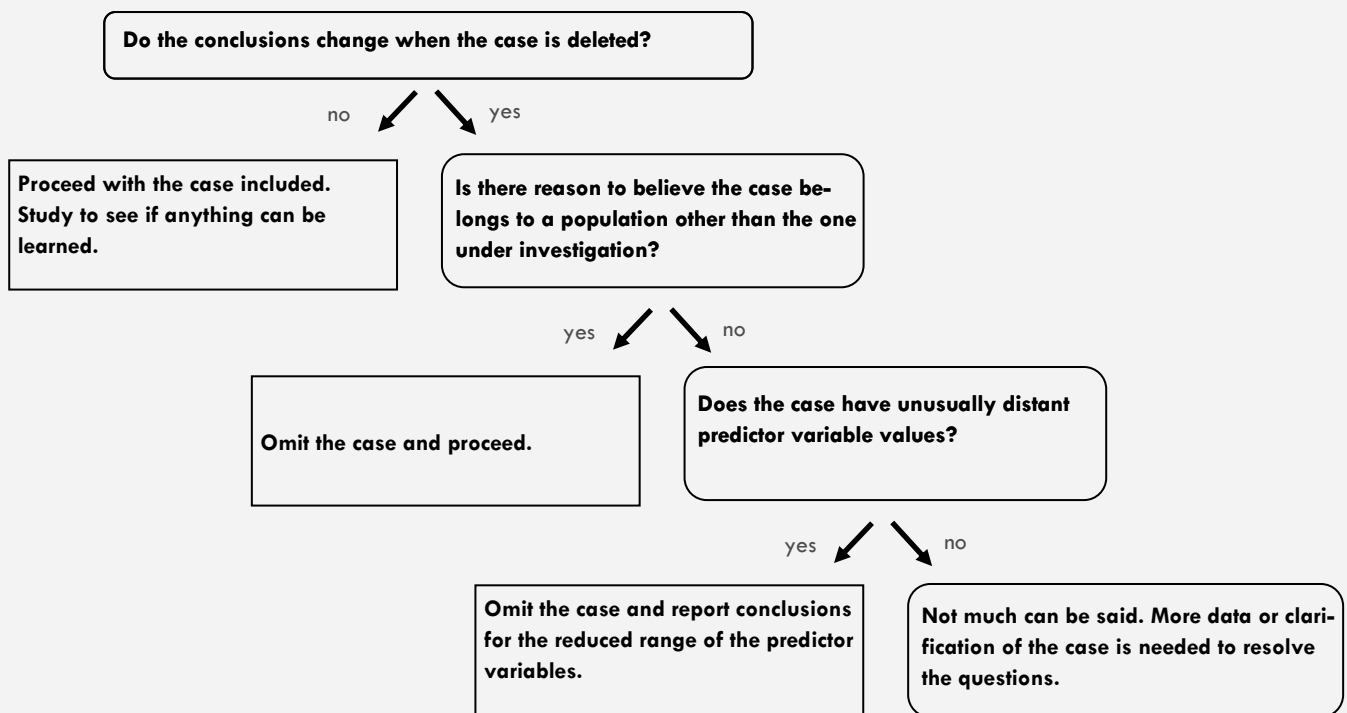
Summary Plot



This plot simultaneously shows Cook's D, Leverage (Hat-Values), and the Studentized Residuals. Each case is represented by a bubble, and the size of the bubble denotes Cook's D. The `id.method = "noteworthy"` labels any points that are concerning large on any of the indicators. ID 204 keeps showing up over and over again. This person is likely someone we should be concerned about.

Remedy for Unusual Cases

A Strategy for Dealing with Unusual Cases



Refit Model 3 Without Case 204

Since case 204 seems to be problematic, let's drop that case and refit the model to determine if the results substantially change.

```
sentence_drop <- filter(sentence, id != 204)
```

```
mod3_drop <- lm(data=sentence_drop, lnyears ~ primlev0 + primlev02 + nsecond + seclev + seclev2 + nprior +  
priorlev + priorlev2 + black + afro_m)  
ols_regress(mod3_drop)
```

Without #204

Parameter Estimates							
model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
(Intercept)	0.478	0.316		1.512	0.132	-0.146	1.102
primlev0	-0.126	0.114	-0.246	-1.102	0.272	-0.351	0.099
primlev02	0.038	0.010	0.849	3.697	0.000	0.018	0.058
nsecond	0.077	0.020	0.199	3.862	0.000	0.037	0.116
seclev	-0.132	0.063	-0.321	-2.100	0.037	-0.257	-0.008
seclev2	0.024	0.008	0.452	2.892	0.004	0.008	0.040
nprior	0.019	0.036	0.033	0.521	0.603	-0.052	0.089
priorlev	-0.019	0.088	-0.041	-0.215	0.830	-0.192	0.154
priorlev2	0.004	0.012	0.049	0.287	0.774	-0.021	0.028
black	-0.321	0.141	-0.151	-2.278	0.024	-0.599	-0.043
afro_m	0.093	0.040	0.155	2.338	0.020	0.015	0.172

With #204

Parameter Estimates							
model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
(Intercept)	0.506	0.316		1.602	0.111	-0.117	1.129
primlev0	-0.138	0.114	-0.270	-1.211	0.227	-0.363	0.087
primlev02	0.039	0.010	0.876	3.831	0.000	0.019	0.059
nsecond	0.058	0.014	0.208	4.233	0.000	0.031	0.085
seclev	-0.110	0.061	-0.267	-1.812	0.071	-0.230	0.010
seclev2	0.021	0.008	0.409	2.676	0.008	0.006	0.037
nprior	0.021	0.036	0.037	0.588	0.557	-0.049	0.091
priorlev	-0.024	0.088	-0.051	-0.269	0.788	-0.197	0.150
priorlev2	0.004	0.012	0.055	0.325	0.746	-0.020	0.028
black	-0.322	0.141	-0.151	-2.280	0.024	-0.600	-0.044
afro_m	0.092	0.040	0.153	2.306	0.022	0.013	0.171

The results are generally the same when we exclude case 204. The slopes for nsecond and seclev shift a bit, but the primary slope of interest (afro_m) changes very little.

Multicollinearity: Detecting It & Remedies

Often the predictor variables in a multiple linear regression are correlated. Usually, the extent to which the predictors are correlated does not create a problem. However, in some cases, two or more predictors may be very highly correlated, such that one or more variables are nearly redundant (i.e., one x can be very well predicted by another x variables). This is called severe multicollinearity.

Severe multicollinearity among predictors may increase the standard errors, cause the regression slopes to be extremely sensitive to minor model modifications (i.e., adding or removing a predictor), cause the slopes to switch signs, and/or negatively impact statistical power.

Severe multicollinearity can make it difficult to choose the correct predictors to include and determine the precise effect of each predictor in the model; however, multicollinearity doesn't affect the overall fit of the model or produce improper predictions (\hat{y} -hats). So, depending on your goal, even a high degree of multicollinearity may not be an issue.

The Variance Inflation Factor (VIF) is a measure of how much the variances of the estimated regression coefficients (i.e., the square of the standard errors) are inflated as compared to when the predictor variables are not correlated. The VIF can be calculated for each predictor by regressing the predictor on all the other predictors. The VIF is then calculated using the R^2 from the model using the following formula: $1/(1-R^2)$. Larger VIF's indicate higher multicollinearity. There are many rules of thumb for VIF, most people indicate that a VIF of 5 is potentially problematic, and a VIF larger than 10 is indicative of serious problems, although others set a much lower threshold (e.g., 2.5).

Calculate Variance Inflation Factors

vif(mod3)

```
primlev0 primlev02  nsecond  seclev  seclev2  nprior  priorlev priorlev2    black  afro_m
24.697602 26.013774  1.207269 10.774113 11.607362  2.023950 17.919008 14.243353  2.196643  2.198933
```

We have several VIFs over 10 here, but notice that all of the culprits are involved in polynomial specifications. Interaction and higher-order terms are correlated with the main effects because they include the main effect terms.

Explore What Happens if We Center the Polynomial Terms

Center polynomial terms at the mean and refit Model 3

```
sentence_cen <- mutate(sentence,
  primlevm = primlev - mean(primlev),
  primlevm2 = primlevm^2,
  seclevm = seclev - mean(seclev),
  seclevm2 = seclevm^2,
  priorlevm = priorlev - mean(priorlev),
  priorlevm2 = priorlevm^2)

mod3_cen <- lm(data=sentence_cen, lnyears ~ primlevm + primlevm2 + nsecond + seclevm + seclevm2 + nprior +
  priorlevm + priorlevm2 + black + afro_m)
ols_regress(mod3_cen)

vif(mod3_cen)
```


Essential vs. Non-essential Collinearity

New model

Model Summary			
R	0.767	RMSE	0.698
R-Squared	0.588	Coef. Var	61.861
Adj. R-Squared	0.568	MSE	0.487
Pred R-Squared	0.535	MAE	0.528

RMSE: Root Mean Square Error
MSE: Mean Square Error
MAE: Mean Absolute Error

ANOVA					
	Sum of Squares	DF	Mean Square	F	Sig.
Regression	142.610	10	14.261	29.276	0.0000
Residual	99.861	205	0.487		
Total	242.471	215			

Parameter Estimates							
model	Beta	Std. Error	Std. Beta	t	Sig.	lower	upper
(Intercept)	0.789	0.135		5.838	0.000	0.522	1.055
primlevm	0.294	0.028	0.575	10.357	0.000	0.238	0.351
primlevm2	0.039	0.010	0.183	3.831	0.000	0.019	0.059
nsecond	0.058	0.014	0.208	4.233	0.000	0.031	0.085
seclvm	0.036	0.021	0.088	1.725	0.086	-0.005	0.077
seclvm2	0.021	0.008	0.149	2.676	0.008	0.006	0.037
nprior	0.021	0.036	0.037	0.588	0.557	-0.049	0.091
priorlevm	-0.012	0.055	-0.026	-0.221	0.825	-0.121	0.097
priorlevm2	0.004	0.012	0.033	0.325	0.746	-0.020	0.028
black	-0.322	0.141	-0.151	-2.280	0.024	-0.600	-0.044
afro_m	0.092	0.040	0.153	2.306	0.022	0.013	0.171
primlevm	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
primlevm2	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
nsecond	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
seclvm	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
seclvm2	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
nprior	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
priorlevm	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
priorlevm2	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
black	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718
afro_m	1.532294	1.136517	1.207269	1.284127	1.545728	2.023950	7.053718

The overall model is identical to what it was with our initial centering strategy, but the coefficients associated with the first order polynomial terms (e.g. primlevm) change because of the centering. For example, the slope for primlevm is the expected slope at the mean of primlev, where in the original model it was the expected slope when primlev0 was 0.

Notice that in the new model, the VIFs for the polynomial terms have all decreased substantially. This is an important observation — multicollinearity that arises from interaction terms or polynomial terms exist because of scaling. Multicollinearity in this case is called non-essential multicollinearity, meaning that it is not something that you need to be concerned about. As long as the variables involved in an interaction or polynomial specification have a meaningful 0 point, then your slopes are interpretable.

This type of multicollinearity can also arise due to the use of dummy coded indicators if the proportion of people in the reference category is small. See <https://statisticalhorizons.com/multicollinearity> for a nice discussion of this and related issues.

On the other hand, severe multicollinearity that is of concern arises when two different variables are highly correlated, (e.g., an adolescent's age and her stage of maturation). When you find a large VIF in this setting, then important problems can arise in interpreting the slopes and associated standard errors, as was discussed on the previous page.

Original model

Model Summary			
R	0.767	RMSE	0.698
R-Squared	0.588	Coef. Var	61.861
Adj. R-Squared	0.568	MSE	0.487
Pred R-Squared	0.535	MAE	0.528

RMSE: Root Mean Square Error
MSE: Mean Square Error
MAE: Mean Absolute Error

ANOVA					
	Sum of Squares	DF	Mean Square	F	Sig.
Regression	142.610	10	14.261	29.276	0.0000
Residual	99.861	205	0.487		
Total	242.471	215			

Parameter Estimates							
model	Beta	Std. Error	Std. Beta	t	Sig.	lower	upper
(Intercept)	0.506	0.316		1.602	0.111	-0.117	1.129
primlev0	-0.138	0.114	-0.270	-1.211	0.227	-0.363	0.087
primlev02	0.039	0.010	0.876	3.831	0.000	0.019	0.059
nsecond	0.058	0.014	0.208	4.233	0.000	0.031	0.085
seclvm	-0.110	0.061	-0.267	-1.812	0.071	-0.230	0.010
seclvm2	0.021	0.008	0.409	2.676	0.008	0.006	0.037
nprior	0.021	0.036	0.037	0.588	0.557	-0.049	0.091
priorlev	-0.024	0.088	-0.051	-0.269	0.788	-0.197	0.150
priorlev2	0.004	0.012	0.055	0.325	0.746	-0.020	0.028
black	-0.322	0.141	-0.151	-2.280	0.024	-0.600	-0.044
afro_m	0.092	0.040	0.153	2.306	0.022	0.013	0.171

Remedy—Multicollinearity

It is difficult to interpret the partial regression coefficients when two predictor variables are very highly correlated. For example, it may not make much sense to interpret the effect of an increase in x_1 when x_2 is held constant if x_1 and x_2 are highly correlated. Here are a few recommendations when multicollinearity is found to be a problem:

1. Do not be concerned about the non-essential multicollinearity that is related to polynomial or interaction terms, or multicollinearity that results from dummy-coded indicators.
2. If the multicollinearity problems are restricted to control variables, and not key variables of interest, then ignoring the multicollinearity is reasonable as the issue only affects estimates and standard errors of the affected variable(s) and not the other slopes or the overall model.
3. If multicollinearity negatively affects your key variables consider assessing the correlated predictors in separate models.
4. If multicollinearity negatively affects your key variables, consider keeping the most salient predictors and dropping the others that are highly correlated with this salient predictor.
5. If multicollinearity negatively affects your key variables, consider forming a composite index or scale based on the correlated predictors. For example, if you had several measures that assess socio-economic status (e.g., income, education, job prestige), form a composite measure of socio-economic status that combines these three variables. You could create a z-score of all three, ensure that all three are coded in the same direction (e.g., higher means higher socio-economic status), and then take the average to form a scale. Then, you would include the scale as a predictor, rather than the three separate measures.