Example 1

Anni Hong

12/30/2020

Todo:

- reference more papers that uses stepwise regression for variable selection
- explain other types of p-value based variable selection, i.e univariate p-value based selection
- figure out why linearity does not hold when data is shuffled

Example 1, model selection using stepwise regression based on p-values:

Textbook:

Probability and Statistics for Engineers and Scientists (4 edition) Anthony Hayter “Model fitting is performed by finding which subset of the k input variables is required to model the dependent variable y in the best and most succinct manner. The final model that the experimenter uses for inference problems should consist of input variables that each have p-values no larger than 10%, because otherwise some variables should be taken out of the model to simplify it. An important warning when fitting the model is that it is best to remove only one variable from the model at a time. This is because when one variable is removed from the model and the subsequent reduced model is fitted, the p-values of the remaining input variables in the model usually change.”

"Typically, model fitting may be performed in the following manner. **First, an experimenter starts by fitting all k input variables. If all variables are needed in the model, then no reduction is necessary. If one or more input variables has a p-value larger than 10%, the variable with the largest p-value (smallest absolute value of the t-statistic) is removed. The reduced model with k -1 input variables is then fitted, and the process is repeated."**

Hayter directly recommended using stepwise regression to select the best model and then conduct statistical inference on the selected model in *Probability and Statistics for Engineers and Scientists*. In the following paper, we first permute all the rows of the dataset while keeping the outcome variables fixed thus remove any association between the covariates and the outcomes. Then we follow the model fitting procedure outlined in the paper in this shuffled

dataset and demonstrate that statistically significant results can be found on this null dataset. Since the exact procedure is not given in the paper, we adhered to the recommendation of the textbook when ambiguity arose.

On Dec.16th 2020, Science Daily published an article Three pillars of mental health: Good sleep, exercise, raw fruits and veggies (https://www.sciencedaily.com/releases/2020/12/201216094647.htm) summarizing a research conducted by the University of Otago study, claiming that getting good quality sleep, exercising, and eating more raw fruits and vegetables predicts better mental health and well-being in young adults. The research paper titled The Big Three Health Behaviors and Mental Health and Well-Being Among Young Adults: A Cross-Sectional Investigation of Sleep, Exercise, and Diet (https://www.frontiersin.org/articles/10.3389/fpsyg.2020.579205/full#h3) was published in *Frontiers in Psychology*. The following sections follows the procedure described in the *Data Preparation and Analyses* section of the paper.

0. The dataset was read in and processed the same way as the paper detailed in the supplementary material section. And then permuted to break the association between covariates and the outcome variables of interests.

```
outcomes <- c("flourishing_c", "cesd_c")

predictors <- c("sleep_quantity_c", "sleep_quality_c", "activity_c", "ra
w_fv_c", "cooked_fv_c", "fastfood_daily_c", "sweets_daily_c", "soda_dail
y_c")

covariates <- c("age_c", "gender", "ethnicity_cat", "sample", "unemploye
d", "ses_c", "bmi_c", "condition", "medmood", "vitsup", "allergy", "vege
tarian", "alcohol_daily_c", "regsmoke")

set.seed(123)
shuffled <- shuffle(data_shay_pre2, outcomes)

shuffled_full <- generate_full_model_dat(shuffled, base_var = covariate
s, add_var = predictors, outcomes = outcomes)
```

1. “[demographic] covariates were included in the model if they correlated with either the predictors and/or the outcome measures.”

```
res2<-rcorr(as.matrix(as.data.frame(lapply(shuffled, as.numeric))))
corr_significant <- as.data.frame(res2[["P"]][covariates,c(predictors,ou
tcomes)])

covariates_kept <- corr_significant %>%
  select_all() %>%
  filter_all(any_vars(.<= 0.05)) %>% rownames()

fit <- lm(as.formula(paste0('cesd_c ~ - flourishing_c +', paste(covariat
es_kept, collapse = ' + '))), data=shuffled)

covariates_kept_factorized <- names(fit$coefficients)[-1]

model_1_flo <- get_result(shuffled_full, outcomes[1], covariates_kept_fa
ctorized, outcomes[2])
model_1_dep <-get_result(shuffled_full, outcomes[2], covariates_kept_fac
torized, outcomes[1])
```

As the above table has shown, all the demographic covariates correlate with at least one predictor of interest or the outcome variable of interest. Thus all will be kept in the model. Those selected covariates will be used for Model 1 in the paper.

2. "Quadratic factors for the sleep, activity, and diet variables were included to test for any non-linear associations with the outcomes and were retained only when significant.

- It is unclear if all quadratic terms thrown in all at once or through a stepwise procedure. We will use a forward stepwise regression that starts with all the demographic covariates and the health behaviors (sleep_quantity_c, sleep_quality_c, activity_c, raw_fv_c, cooked_fv_c, fastfood_daily_c, sweets_daily_c, soda_daily_c) and end with the starting model plus all the quadratic health behavior terms.

ref for 0.05 significant level when doing stepwise:
https://stats.stackexchange.com/questions/97257/stepwise-regression-in-r-critical-p-value (https://stats.stackexchange.com/questions/97257/stepwise-regression-in-r-critical-p-value)

```
quad_selector <- selector_factory()
kept_flo_2 <- quad_selector(shuffled_full, base_var = covariates_kept, a
dd_var = predictors, outcome = outcomes[1], remove = outcomes[2])

kept_dep_2 <- quad_selector(shuffled_full, base_var = covariates_kept, a
dd_var = predictors, outcome = outcomes[2], remove = outcomes[1])

model_2_flo <- get_result(shuffled_full, outcomes[1], kept_flo_2, outcom
es[2])
model_2_dep <-get_result(shuffled_full, outcomes[2], kept_dep_2, outcome
s[1])
```

model 3 added the significant two-way interaction terms among the health behaviors while keeping significant terms from step 2 in the model.

```
int_selector <- selector_factory(expander = generate_interaction_terms)
kept_flo_3 <- int_selector(shuffled_full, base_var = kept_flo_2, add_var
= predictors, outcome = outcomes[1], remove = outcomes[2])

kept_dep_3 <- int_selector(shuffled_full, base_var = kept_dep_2, add_var
= predictors, outcome = outcomes[2], remove = outcomes[1])

model_3_flo <- get_result(shuffled_full, outcomes[1], kept_flo_3, outcom
es[2])
model_3_dep <-get_result(shuffled_full, outcomes[2], kept_dep_3, outcome
s[1])
```

Through the model selection process described in the research, we discovered many “significant” predictors in the shuffled dataset where no association should be found. Additionally, the p-values for significance were not adjusted for multiple testing thus further invalidate the inference.

remedy for post selection inference

Aware of the replication crisis in Psychology, the researchers write, “We also used 10-fold cross-validation to determine whether any interaction terms would be useful for predicting out-of-sample, above and beyond the no-interaction model. Cross-validation involves splitting data into several subsets or “folds” and then repeatedly fitting the model to all but one fold and testing the model on the leftover fold (Koul et al., 2018)."
Through their cross-validation procedure, the researchers concluded that none of the interaction terms they included in the model helps with out-of-sample prediction.

However, they did not subject the chosen quadratic terms to the same procedure. Moreover, the cross-validation was done on the *same* dataset that the 2-way interaction terms were chosen thus the cv errors are biased estimates of the true test error. The following section demonstrates a remedy for the model selection procedure used in the paper through sample splitting.

1. The dataset is splitted into training (n = 800) and testing(n = 311) sets.

```
split <- sample_splitting(shuffled_full, 0.8)
shuffled_train <- split[[1]]
shuffled_test <- split[[2]]
```

step 2 was also repeated on the training set, slightly different quadratic terms were selected.

Now we extract the set of selected variables from the above procedure and run the same model on the testing set which remained untouched in the selection process. As expected, the significant quadratic terms ceased to be significant.

```
library(stargazer)
stargazer(model_2_dep, model_2_dep_test, type = "html",  title = "Predic
ting Depressive Symptoms: null data", column.labels = c("train", "test"
), dep.var.labels = c("depression", "depression"))
```

Predicting Depressive Symptoms: null data

	Dependent variable:	
	depression	
	train	test
	(1)	(2)
age_c	0.312	0.188
	(0.277)	(0.591)
genderGenderDiverse	4.307	3.562
	(3.802)	(6.032)
genderMale	-1.394	3.279
	(1.060)	(2.245)
ethnicity_catAsian	3.367**	-1.695
	(1.671)	(4.057)
ethnicity_catBlack	-0.123	-1.250
	(2.030)	(4.312)
ethnicity_catHispanic	1.262	3.257
	(2.465)	(3.876)
ethnicity_catOther	1.397	0.992

	(1.176)	(2.909)
sample1	-0.416	-1.736
	(1.542)	(3.314)
unemployedUnemployed	0.754	-0.206
	(1.523)	(3.953)
ses_c	0.699*	0.015
	(0.372)	(0.852)
bmi_c	-0.067	-0.077
	(0.075)	(0.177)
conditionHasnohealthconditions	0.867	-0.845
	(1.035)	(2.163)
medmoodYES	1.025	1.141
	(1.181)	(2.337)
vitsup	-0.320	-3.437***
	(0.567)	(1.129)
allergyNoFoodAllergy	-0.256	-0.334
	(1.256)	(2.349)
vegetarianEatssomemeats	0.416	8.514*
	(1.737)	(4.684)
alcohol_daily_c	0.778**	0.786
	(0.393)	(0.898)
regsmoke1	-0.322	0.458
	(1.625)	(3.389)
sleep_quantity_c	-0.209	-0.454
	(0.334)	(0.719)
sleep_quality_c	-0.856*	-0.321
	(0.503)	(1.061)
activity_c	0.769***	0.055
	(0.250)	(0.538)
raw_fv_c	-0.648**	-0.460
	(0.290)	(0.660)
cooked_fv_c	-0.252	2.410**
	(0.455)	(1.086)
fastfood_daily_c	1.047	2.499
	(1.831)	(5.315)
sweets_daily_c	-0.579	-0.669
	(0.499)	(1.116)
soda_daily_c	0.116	-0.072
	(0.486)	(0.940)
I(sleep_quality_c2)	0.778**	0.259

	(0.369)	(0.778)
I(fastfood_daily_c2)	-1.199*	2.865
	(0.670)	(12.683)
Constant	-1.727	-4.598
	(2.340)	(5.828)
Observations	889	222
R^2	0.050	0.112
Adjusted R^2	0.019	-0.017
Residual Std. Error	12.913 (df = 860)	12.883 (df = 193)
F Statistic	1.625** (df = 28; 860)	0.866 (df = 28; 193)
Note:		*p<0.1;* ***p<0.05;*** p<0.01

```
stargazer(model_2_flo, model_2_flo_test, type = "html", title = "Predict
ing Flourishing: null data", column.labels = c("train", "test"), dep.va
r.labels = c("flourishing", "flourishing"))
```

Predicting Flourishing: null data

	Dependent variable:	
	flourishing	
	train	test
	(1)	(2)
age_c	-0.046*	-0.060
	(0.026)	(0.054)
genderGenderDiverse	-0.345	0.267
	(0.355)	(0.554)
genderMale	0.122	-0.227
	(0.099)	(0.204)
ethnicity_catAsian	-0.346**	0.288
	(0.156)	(0.368)
ethnicity_catBlack	-0.289	-0.340
	(0.190)	(0.397)
ethnicity_catHispanic	0.063	0.086
	(0.230)	(0.357)
ethnicity_catOther	-0.218**	0.169
	(0.110)	(0.267)
sample1	0.160	0.093
	(0.144)	(0.305)
unemployedUnemployed	-0.133	0.170
	(0.142)	(0.364)
ses_c	-0.049	0.027

	(0.035)	(0.079)
bmi_c	-0.001	0.020
	(0.007)	(0.016)
conditionHasnohealthconditions	-0.102	0.199
	(0.097)	(0.198)
medmoodYES	-0.070	-0.121
	(0.110)	(0.215)
vitsup	0.003	0.232**
	(0.053)	(0.104)
allergyNoFoodAllergy	0.049	0.032
	(0.117)	(0.216)
vegetarianEatssomemeats	0.021	-0.246
	(0.162)	(0.429)
alcohol_daily_c	-0.048	-0.074
	(0.037)	(0.083)
regsmoke1	0.219	-0.021
	(0.152)	(0.311)
sleep_quantity_c	0.059*	0.050
	(0.031)	(0.066)
sleep_quality_c	0.006	0.108
	(0.046)	(0.096)
activity_c	-0.030	-0.021
	(0.023)	(0.049)
raw_fv_c	0.023	0.017
	(0.027)	(0.060)
cooked_fv_c	0.028	-0.122
	(0.043)	(0.100)
fastfood_daily_c	-0.196	0.530
	(0.171)	(0.490)
sweets_daily_c	0.039	0.010
	(0.047)	(0.102)
soda_daily_c	-0.003	0.027
	(0.045)	(0.087)
I(fastfood_daily_c2)	0.122*	-1.322
	(0.063)	(1.168)
Constant	-0.050	-0.006
	(0.217)	(0.537)
Observations	889	222
R^2	0.039	0.088
Adjusted R^2	0.009	-0.039

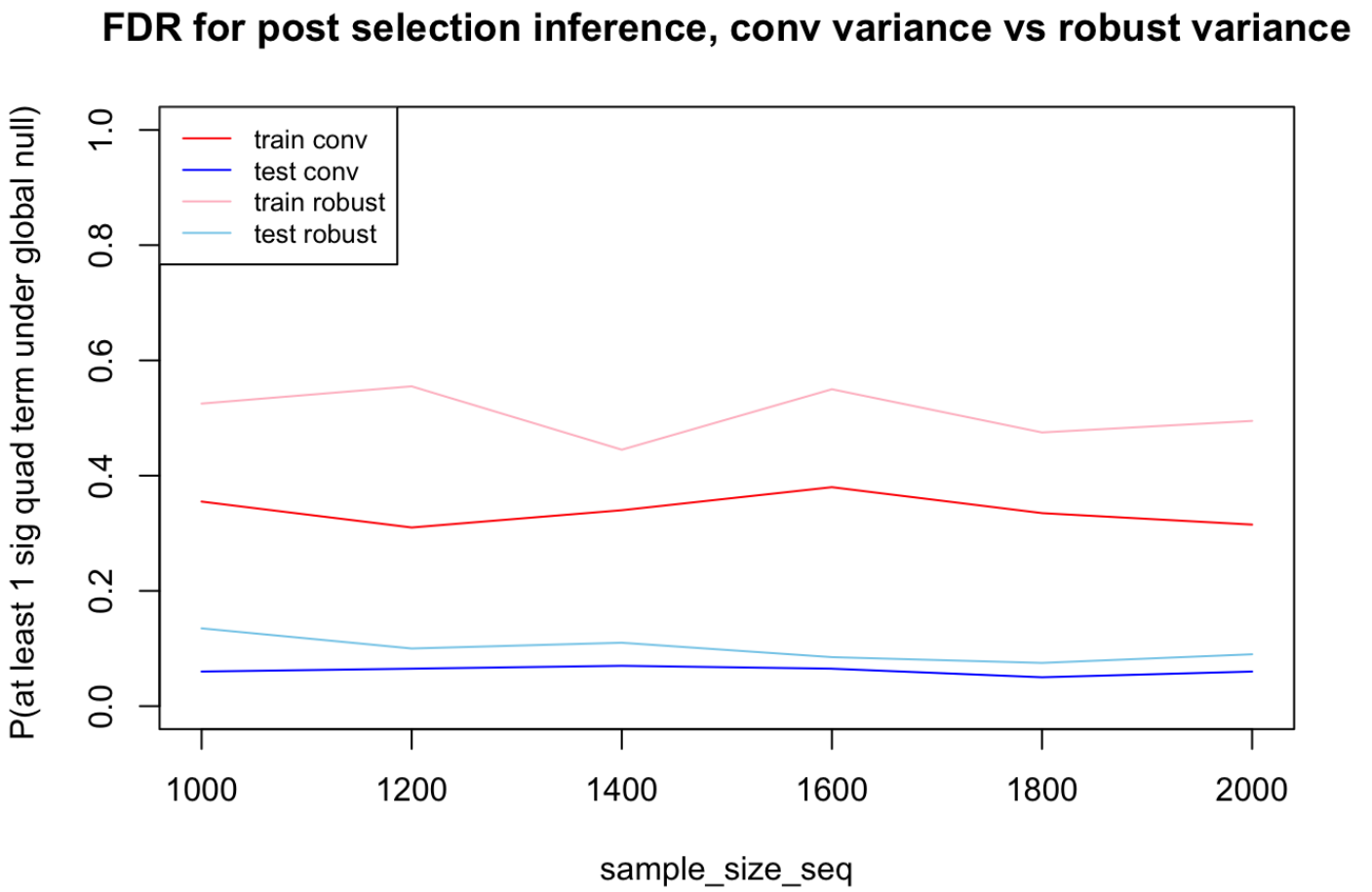
Residual Std. Error	1.207 (df = 861)	1.187 (df = 194)
F Statistic	1.301 (df = 27; 861)	0.692 (df = 27; 194)

Note: *p<0.1;* ***p<0.05;*** p<0.01

```
#only setting quadratic terms
bootstrap_process <- function(data, B, outcome_idx, to_shuffle = T, m=nr
ow(data)) {
  train_res <- list()
  test_res <- list()
  remove_idx <- setdiff(c(1,2),outcome_idx)
  data <- generate_full_model_dat(data, covariates, predictors, outcome
s)
  for (i in 1:B) {
    # #1. bootstrap
    # dat <- bootstrap_sample(data, m)
    # #2. shuffle
    # if (to_shuffle) {
    #   dat <- shuffle(dat, outcomes)
    # }

    # step 1 and 2 combined:
    dat <- cbind(bootstrap_sample(data[,-which(names(data) %in% outcome
s)], m), bootstrap_sample(data[,outcomes], m))

    #3. sample_splitting
    split <- sample_splitting(dat)
    train <- split[[1]]
    test <- split[[2]]

    selector <- selector_factory(expander = generate_quad_terms)
    train_kept <- selector(train, base_var = covariates, add_var = predi
ctors, outcome = outcomes[outcome_idx], remove = outcomes[remove_idx])

    fit_train <- get_result(train, outcomes[outcome_idx], train_kept, ou
tcomes[remove_idx])
    # print(summary(fit_train))
    fit_test <- get_result(test, outcomes[outcome_idx], train_kept, outc
omes[remove_idx])
    # print(summary(fit_test))
    train_res[[i]] <- fit_train
    test_res[[i]] <- fit_test
  }
  return(list(train_res, test_res))
}
```

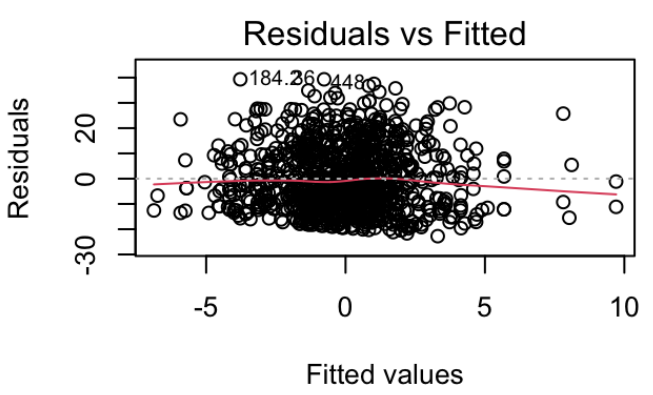
```
train_prop <- c()
test_prop <- c()
train_prop_sandwich <- c()
test_prop_sandwich <- c()
sample_size_seq <- seq(from=1000,to=2000, by=200)
for (size in sample_size_seq){
  r <- bootstrap_process(data=data_shay_pre2, B=200, outcome_idx=2, m=si
ze)
  train_res <- r[[1]]
  test_res <- r[[2]]

  #lm variance
  train_sig <- purrr::map(train_res,get_num_sig_quad_term,robust=F)
  test_sig <- purrr::map(test_res,get_num_sig_quad_term,robust=F)
  train_prop <- c(train_prop, mean(train_sig > 0))
  test_prop <- c(test_prop, mean(test_sig > 0))

  #robust variance
  train_sig_sandwich <- purrr::map(train_res,get_num_sig_quad_term,robus
t=T)
  test_sig_sandwich <- purrr::map(test_res,get_num_sig_quad_term,robust=
T)
  train_prop_sandwich <- c(train_prop_sandwich, mean(train_sig_sandwich
 > 0))
  test_prop_sandwich <- c(test_prop_sandwich, mean(test_sig_sandwich > 0
))
}
```

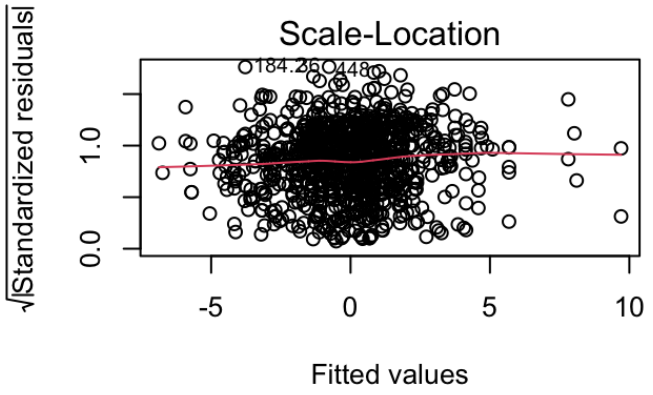
```
#par(mfcol=c(2,1))
plot(sample_size_seq, test_prop, type = "l", col="blue", ylim = c(0,1),
 ylab = "P(at least 1 sig quad term under global null) ", main = "FDR fo
r post selection inference, conv variance vs robust variance")
lines(sample_size_seq,train_prop, col="red")
lines(sample_size_seq,test_prop_sandwich, col="skyblue")
lines(sample_size_seq,train_prop_sandwich, col="pink")
legend("topleft", legend=c("train conv", "test conv", "train robust", "t
est robust"),col=c("red", "blue", "pink", "skyblue"), lty=1:1, cex=0.8)
```

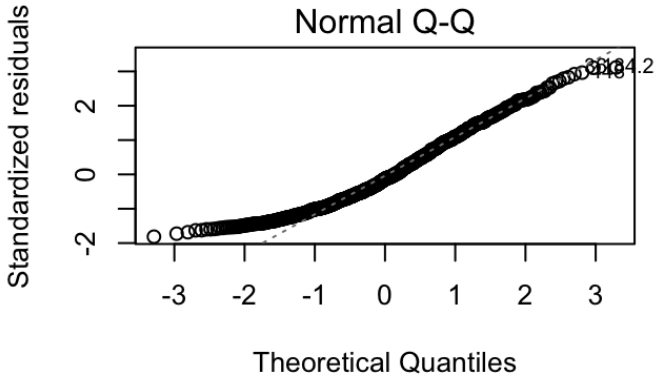


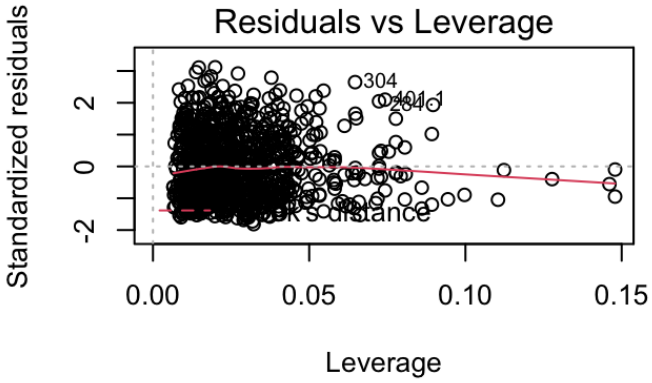
As expected, the false discovery proportion is much higher in the training set where the quadratic terms were selected on. However, the false discovery proportion is also high when the significance level was determined by the robust variance instead the conventional linear model variance. The following diagnostic plots elucidate the reason behind such phenomenon.

```
par(mfcol=c(2,2))
plot(train_res[[sample(length(train_res),1)]])
```

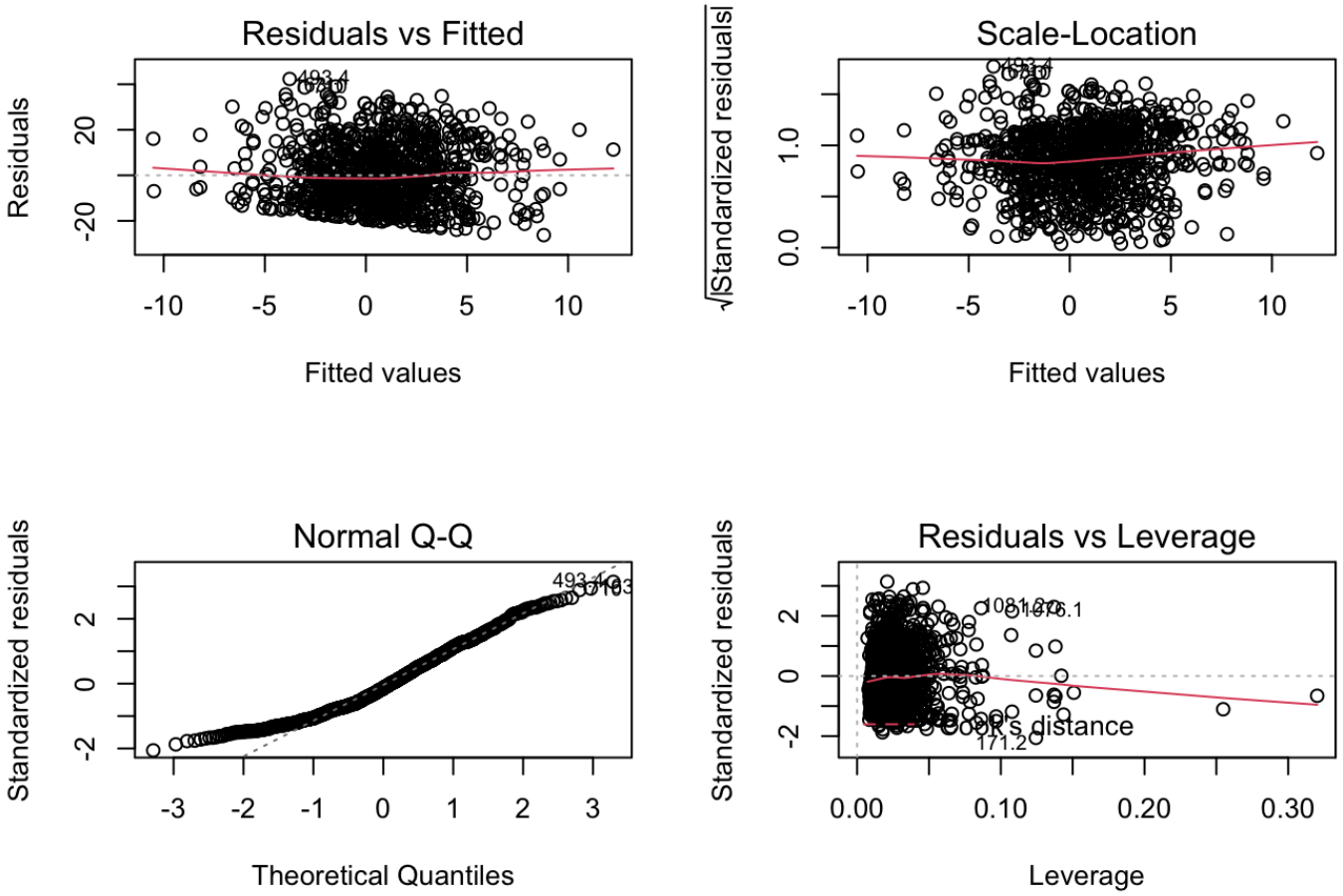






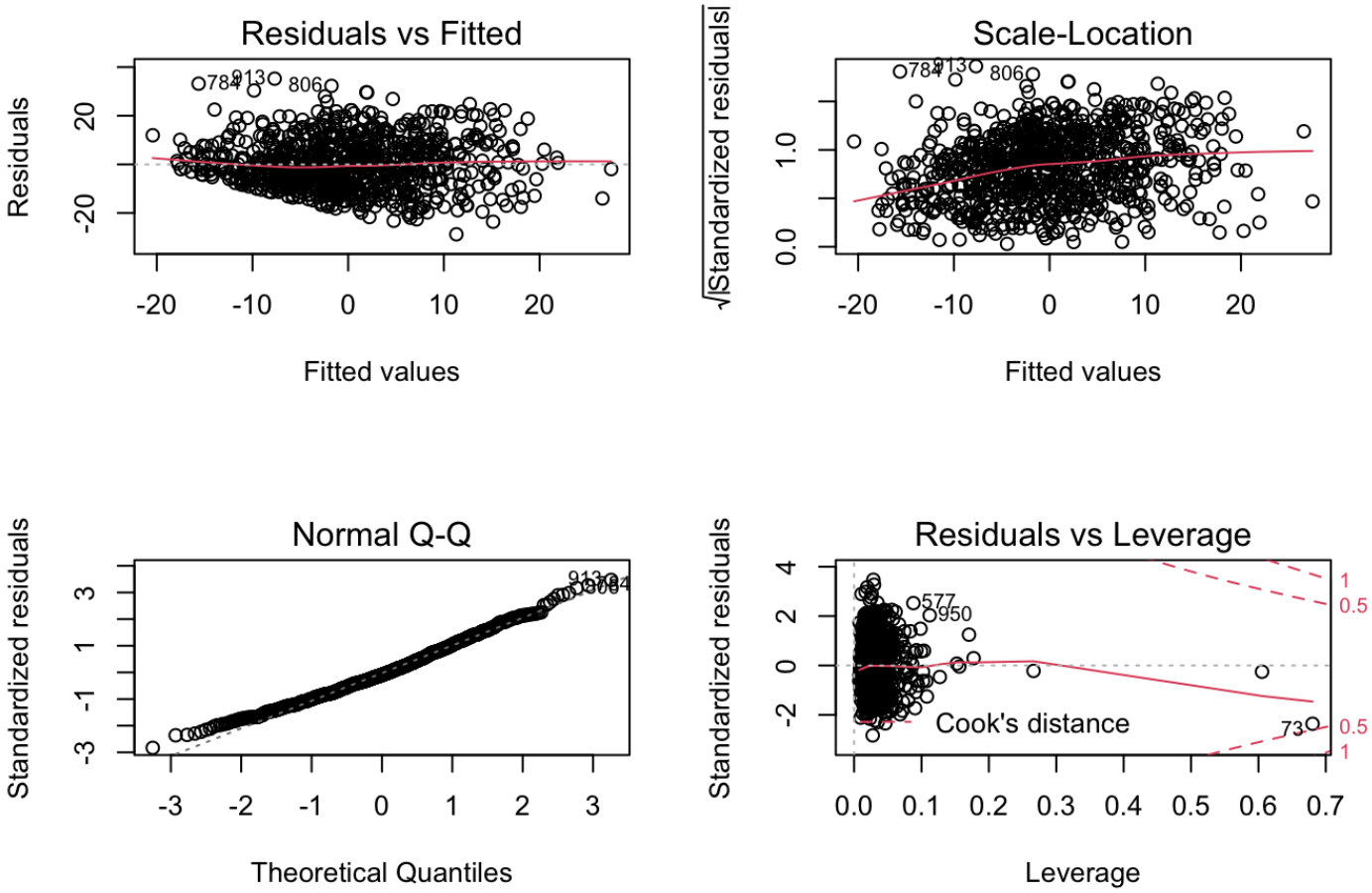



```
plot(test_res[[sample(length(test_res),1)]])
```



The residual has lower variance at more extreme values of of the fitted values, but larger variance closer to the average of the fitted values. This type of heteroskedasticity causes the conventional standard error to bias downward. > The attached note (http://econ.lse.ac.uk/staff/spischke/mhe/josh/Notes%20on%20conv%20std%20error.pdf) describes the mechanics, and gives conditions for the direction of the bias. Basically, conventional standard errors are too big whenever covariate values far from the mean of the covariate distribution are associated with lower variance residuals (so small residuals for small and big values of x, and large residuals in the middle of the x range). We think this is empirically not the common case but it might happen. The leading case is probably that residual variance goes up with the value of x (true for example in the returns to schooling example: earnings are more variable for those with more schooling). In this case, conventional standard errors will tend to be “about right” or too small as the discussion in 8.1 suggests. reference (http://www.mostlyharmlesseconometrics.com/2010/12/heteroskedasticity-and-standard-errors-big-and-small/)

```
#3. sample_splitting
dat <- generate_full_model_dat(data_shay_pre2, covariates, predictors, o
utcomes)
split <- sample_splitting(dat, 0.8)
train <- split$train
test <- split$test
outcome_idx <- 2
remove_idx <- 1

selector <- selector_factory(expander = generate_quad_terms)
train_kept <- selector(train, base_var = covariates, add_var = predictor
s, outcome = outcomes[outcome_idx], remove = outcomes[remove_idx])

fit_train <- get_result(train, outcomes[outcome_idx], train_kept, outcom
es[remove_idx])
fit_test <- get_result(test, outcomes[outcome_idx], train_kept, outcomes
[remove_idx])
require(sandwich)
fit_train_sand <- lmtest::coeftest(fit_train, vcov=sandwich)
fit_test_sand <- lmtest::coeftest(fit_test, vcov=sandwich)
par(mfcol=c(2,2))
plot(fit_train)
```




```
library(stargazer)
stargazer(fit_train, fit_test, type = "html",  title = "Predicting Depre
ssive Symptoms real data", column.labels = c("train", "test"), dep.var.l
abels = c("depression", "depression"))
```

Predicting Depressive Symptoms real data

	Dependent variable:	
	depression	
	train	test
	(1)	(2)
age_c	-0.832***	0.652
	(0.221)	(0.470)
genderGenderDiverse	3.077	6.734
	(2.589)	(11.113)
genderMale	-1.298	-0.485
	(0.841)	(1.743)
ethnicity_catAsian	2.139	0.825
	(1.363)	(2.909)

ethnicity_catBlack	0.087	0.474
	(1.661)	(3.277)
ethnicity_catHispanic	4.898***	-2.454
	(1.882)	(3.517)
ethnicity_catOther	1.006	0.390
	(0.966)	(1.979)
sample1	1.819	-4.141
	(1.246)	(2.510)
unemployedUnemployed	1.422	2.163
	(1.264)	(2.505)
ses_c	-2.041***	-3.475***
	(0.305)	(0.627)
bmi_c	0.094	-0.018
	(0.062)	(0.128)
conditionHasnohealthconditions	-3.145***	-7.341***
	(0.839)	(1.605)
medmoodYES	3.774***	4.140**
	(0.920)	(2.099)
vitsup	-0.003	1.081
	(0.444)	(0.940)
allergyNoFoodAllergy	-0.909	-0.980
	(1.013)	(1.845)
vegetarianEatssomemeats	-1.430	4.491
	(1.458)	(2.925)
alcohol_daily_c	0.074	-0.815
	(0.314)	(0.702)
regsmoke1	-0.185	3.096
	(1.313)	(2.887)
sleep_quantity_c	-0.986***	-0.650
	(0.295)	(0.602)
sleep_quality_c	-3.819***	-2.724***
	(0.395)	(0.854)
activity_c	-0.513**	-0.277
	(0.201)	(0.431)
raw_fv_c	-0.191	-0.559
	(0.241)	(0.455)
cooked_fv_c	0.339	2.331***
	(0.368)	(0.802)
fastfood_daily_c	2.856*	1.935
	(1.493)	(3.948)

sweets_daily_c	0.594	1.288
	(0.408)	(0.794)
soda_daily_c	0.107	0.119
	(0.376)	(0.823)
I(sleep_quantity_c2)	0.180***	0.566*
	(0.060)	(0.309)
I(fastfood_daily_c2)	-0.822	-15.540**
	(0.540)	(7.622)
Constant	1.529	3.819
	(1.909)	(3.977)
Observations	889	222
R^2	0.372	0.513
Adjusted R^2	0.352	0.442
Residual Std. Error	10.297 (df = 860)	10.282 (df = 193)
F Statistic	18.231*** (df = 28; 860)	7.262*** (df = 28; 193)
Note:		*p<0.1;* ***p<0.05;*** p<0.01

```
stargazer(fit_train_sand, fit_test_sand, type = "html",  title = "Predic
ting Depressive Symptoms real data robust", column.labels = c("train",
"test"), dep.var.labels = c("depression", "depression"))
```

Predicting Depressive Symptoms real data robust

	Dependent variable:	
	depression	
	train	test
	(1)	(2)
age_c	-0.832***	0.652
	(0.226)	(0.429)
genderGenderDiverse	3.077	6.734
	(2.667)	(4.313)
genderMale	-1.298	-0.485
	(0.844)	(1.556)
ethnicity_catAsian	2.139	0.825
	(1.314)	(2.342)
ethnicity_catBlack	0.087	0.474
	(1.697)	(3.583)
ethnicity_catHispanic	4.898***	-2.454
	(1.793)	(3.462)
ethnicity_catOther	1.006	0.390
	(0.993)	(1.844)

sample1	1.819	-4.141*
	(1.230)	(2.367)
unemployedUnemployed	1.422	2.163
	(1.193)	(2.250)
ses_c	-2.041***	-3.475***
	(0.298)	(0.549)
bmi_c	0.094	-0.018
	(0.064)	(0.119)
conditionHasnohealthconditions	-3.145***	-7.341***
	(0.865)	(1.623)
medmoodYES	3.774***	4.140*
	(0.973)	(2.238)
vitsup	-0.003	1.081
	(0.489)	(0.968)
allergyNoFoodAllergy	-0.909	-0.980
	(1.064)	(1.790)
vegetarianEatssomemeats	-1.430	4.491
	(1.409)	(2.740)
alcohol_daily_c	0.074	-0.815
	(0.293)	(0.540)
regsmoke1	-0.185	3.096
	(1.413)	(3.041)
sleep_quantity_c	-0.986***	-0.650
	(0.331)	(0.586)
sleep_quality_c	-3.819***	-2.724***
	(0.383)	(0.825)
activity_c	-0.513***	-0.277
	(0.194)	(0.402)
raw_fv_c	-0.191	-0.559
	(0.216)	(0.394)
cooked_fv_c	0.339	2.331***
	(0.343)	(0.742)
fastfood_daily_c	2.856**	1.935
	(1.448)	(3.611)
sweets_daily_c	0.594	1.288**
	(0.409)	(0.603)
soda_daily_c	0.107	0.119
	(0.407)	(0.898)
I(sleep_quantity_c2)	0.180**	0.566*
	(0.071)	(0.288)

I(fastfood_daily_c2)	-0.822**	-15.540*
	(0.386)	(8.948)
Constant	1.529	3.819
	(1.927)	(3.936)
Note:	*p<0.1;* ***p<0.05;*** p<0.01	

```
# stargazer(flo, flo_test, type = "html", title = "Predicting Flourishin
g real data", column.labels = c("train", "test"), dep.var.labels = c("fl
ourishing", "flourishing"))
```