ProbSet 5, February 13

Gideon Gordon gideongordon2029@u.northwestern.edu

Due: February 6, 2026

Submission: https://canvas.northwestern.edu/courses/245562/assignments/1687750

0.1 Problem 1

1a. In your own words, explain: 1. The difference between residuals (e_i) and modeling errors ($_i$). A residual is what you get even when you correctly estimate the expected value — because the expected value, while it can be exactly correct as an expectation with an infinite population, does not match any particular outcome of the formula. So a residual is the distance between a particular outcome and the expected value at the population level. In an infinite population the expectation is exactly correct, but as long as the CEF allows for any variation at all, a residual will still exist. Because the residual is a gap between observed data and estimates, it is impossible to get rid of completely even with a perfect population-level dataset. When we write e_i we're looking for the residual at a given value of X; the gap between the value y_i we observe when $x = i$ and the value of y_i predicted by our model for $x = i$.

The modeling error $_i$ is the error that represents the gap between our estimates and the real distribution of data. We know what the residuals are for a given model, since they come from two quantities we know (the observed information and our estimate). We really don't know for sure what our modeling error is — what the gap is, for example, between our estimated

CEF and the actual CEF. When we say $_i$ we're talking about the gap between the y_i we estimate from our data and the y_i we would expect, if we had the actual CEF available to us. But we don't. We only know what we can estimate from the data, so we can't really know what the modeling error is and can never get rid of it. It haunts us, like a heart beating under the floorboards in a Poe short story, ever present but impossible to find.

2. Why the sum of residuals in OLS regression equals zero when an intercept is included. So, when an intercept is *not* included in an OLS regression, we get a regression line which minimizes residuals, but has to pass through the origin (because we can't include a constant term to give us a value of y when $x = 0$). Since we're stuck running the regression through the origin, we'll find a regression line with a slope to minimize residuals. Since the regression is farther to one side of the true BLP than to the other, the residuals above or below the line will be bigger; there will be an imbalance in the residuals. This will make the *sum* of residuals (not necessarily the sum of squared residuals) lean to one side or the other, since the negative residuals will not counterbalance the positive residuals.

So let's say the actual BLP is $y = x + 5$, with observations $(1, 6)$, $(1, 1)$, $(1, 11)$, $(2, 8)$, $(2, 6)$, $(3, 8)$, $(3, 1)$, $(3, 17)$. If we just drew a line without an intercept, we would note that the estimated expected values at y_1, y_2 and y_3 are respectively $y_1 = 6$, $y_2 = 7$, and $y_3 = 8$. But the best we can do, without an intercept available, is note that an increase of 1 in x produces an increase of 1 in y. The residuals resulting from this estimate would be, respectively, 5, 0, 10, 6, 4, 5, -4, and 14, summing to 40.

Adding an intercept fixes this imbalance. It removes the restraint of having to go through the origin, leaving us free to find the true BLP, where the negative residuals balance the positive ones.

If we add +5 to the BLP estimate above, we can estimate the real expected values, and our residuals follow suit; residuals would now be, respectively, 0, -5, 5, 1, -1, 0, -9, and 9, and

they would sum to 0. Note that this does not remove the sum of *squared* residuals, since squaring our residuals gets them to all be positive. So our squared residuals for $y = x + 5$ would be 0, 25, 25, 1, 1, 0, 81, and 81, with a sum of 216. On the other hand our squared residuals for the previous line, $y = x$, would be 25, 0, 100, 36, 16, 25, 16, and 196, summing to 414. So we have reduced our sum of squared residuals by adding an intercept, but not to zero. It would be very unlikely to reduce our squared residuals to zero in any case.

3. How R^2 measures model fit and what it represents. R^2 is basically a measurement of how well a model fits, relative to a much worse model where we don't bother with any fancy OLS math and just assume that the BLP is a constant equal to the mean value of y.

Let's go back to $y = x+5$. If we want to know how much better this BLP is than an estimate not involving scary numbers, we can shrug and model this as $y = 56/7 = 8$ (56 is the sum of y-values, and we have 7 observations). Then we can do residuals for this very crappy model, which will be -2, -7, 3, 0, -2, 0, -9, 9, summing to -8. Our squared residuals then will be 4, 49, 9, 0, 4, 0, 81, and 81, summing to 228.

So yeah, our model with a constant has bigger squared distances between observations and the line (228) than our model with an actual formula (216). R^2 is how we can tell exactly *how* much worse the constant model is. $R^2 = 1\ \frac{RSS}{TSS}$, where RSS is the sum of squared residuals (of an actual model) and TSS is the total squared sums (of distances from our crappy constant). In this case, $R^2 = 1\ \frac{216}{228}$, which is 1 .94 which is 0.06. So, in this case, my BLP $y = x + 5$ does not improve that much on just drawing a straight line, but it does improve on it a little. Maybe there's some variation in y that isn't explained by x alone; further research needed; please give me grant funding.

1b. Using the Hibbs election data:

```r
# Load data
library(rosdata)
library(tidyr)
```

```
Warning: package 'tidyr' was built under R version 4.5.2
```

```r
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.5.2
```

```r
data("hibbs")


# Fit the models
model_intercept <- lm(vote ~ 1, data = hibbs)  # Intercept only
model_econ <- lm(vote ~ growth, data = hibbs)   # With growth
```

Questions: 1. Verify that $e_i = 0$ for both models.

```r
sum(residuals(model_intercept))
```

```
[1] -1.554312e-15
```

```r
sum(residuals(model_econ))
```

```
[1] 3.996803e-15
```

I mean basically yeah. Close enough for government work. Especially under our present government.

2. Calculate R^2 using the formula $R^2 = 1 \ \frac{RSS}{TSS}$.

```r
1 - sum((residuals(model_econ)^2))/sum((residuals(model_intercept)^2))
```

```
[1] 0.5798462
```

Yep, we got an R2 of 0.579.

3. Compare your calculation with the `summary()` output.

```
summary(model_econ)$r.squared
```

```
[1] 0.5798462
```

Precisely identical.

1c. Limitations of R^2: 1. What happens to R^2 when you add more variables to a model, even irrelevant ones?

No matter what happens, when you add variables, R^2 goes up. So for example, we can create `model_superstitious`, which includes the numeric value of the first letter in the incumbent's surname as a separate variable. This is because I believe that names have a great magical influence over our fates.

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
hibbs <- mutate(hibbs, first_letter_incumbent = substr(inc_party_candidate, start = 1, s
hibbs <- mutate(hibbs, first_letter_incumbent = match(first_letter_incumbent, LETTERS))

model_superstitious <- lm(vote ~ growth + first_letter_incumbent, data = hibbs)
summary(model_superstitious)$r.squared
```

```
[1] 0.5945585
```

Hey look! I went from an R^2 of 0.57 to an R^2 of 0.59! 2% more of the variation in vote totals is explained if we control for the mystical effect of the first letter of a surname on the fate of a candidate. I would like my Nobel Prize in Political Science now. (Is that not a thing? The spirits are telling me that's not a thing.)

2. Why might a high R^2 not indicate a good model?

As the example of `model_superstitious` indicates, you can just add variables ad nauseam to increase R^2. Additional variables will most likely capture random variation, even by pure chance. Adjusted R^2 helps with this by only increasing R^2 when a new variable contributes more than random chance... and penalizes R^2 otherwise.

```
summary(model_superstitious)$adj.r.squared
```

```
[1] 0.5321829
```

Darn, guess I won't be getting a Nobel in political science today.

The main thing is, maximizing R^2 doesn't actually tell you anything about whether a relationship is meaningful, what direction things go, whether a relationship is statistically different from nonexistent. If you throw covariate pasta at the scatterplot, some of it will stick. That doesn't actually mean anything for understanding the world we live in.

0.2 Problem 2

2a. 1. Write down the regression model in matrix form and explain each symbol.

$$y = X +$$

- y is the vector of values of our outcome of interest.

- X is our matrix of covariates (with the first row usually a row of 1s for our intercept). Each column is a list of values of covariates in different cases.
- is our vector of coefficients, which we can estimate with the equation $_{OLS} = (\mathbf{X}^T\mathbf{X})^1(\mathbf{X}^T\mathbf{Y})$.
- is our error term. It's how much random nonsense gets thrown in between the amount we predict with our coefficients and the amount we get on y. There is no escaping it.

2b. Create a multicollinear scenario:

```
library(car)
```

```
Warning: package 'car' was built under R version 4.5.2
```

```
Loading required package: carData
```

```

Attaching package: 'car'
```

```
The following object is masked from 'package:dplyr':

    recode
```

```
# Create multicollinear data
set.seed(123)
n <- 100
x1 <- rnorm(n)
x2 <- 0.95*x1 + rnorm(n, sd = 0.1)  # Highly correlated with x1
x3 <- rnorm(n)
y <- 2 + 1.5*x1 + 0.8*x3 + rnorm(n)

# Fit models
```

```r
model_collinear <- lm(y ~ x1 + x2 + x3)
summary(model_collinear)
```

```
Call:
lm(formula = y ~ x1 + x2 + x3)

Residuals:
     Min       1Q   Median       3Q      Max
-2.49138 -0.65392  0.05664  0.67033  2.53210

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.9807     0.1073  18.452  < 2e-16 ***
x1            1.0055     1.0410   0.966    0.337
x2            0.4622     1.0946   0.422    0.674
x3            0.7426     0.1122   6.617 2.09e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.052 on 96 degrees of freedom
Multiple R-squared:  0.6503,    Adjusted R-squared:  0.6394
F-statistic: 59.52 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
# Check variance inflation factors (VIF)
vif(model_collinear)
```

```
       x1        x2        x3
80.852638 80.779423  1.017576
```

Questions: 1. What happens to the standard errors when multicollinearity is present? They get way bigger, because we have two variables basically splitting up the same explanatory task so neither one is given a full chance to deal with it. Also, even though `x1` actually does contribute to `y` and `x2` does not, there's no way for the formula to tell that, so both end up a mess.

2. How do the VIF values indicate multicollinearity? To level with you, I don't think we talked about the VIF in class. I might be wrong. But some Googling reveals that the VIF is given by the ratio of the variance of the model with all the parameters, to the variance given with a model using *only* a certain covariate. So basically, does the variance of this estimate get less absurd if we leave out all other parameters, and if so, how much less absurd.

We don't know *which* other parameters might be covariates, but we can at least tell that a parameter has at least one covariate. And in this case we see there are only two parameters with crazy high VIFs, so we have a pretty good sense.

3. What are the practical implications for interpreting coefficients in the presence of multicollinearity? When we have perfect multicollinearity, the regression will just fall apart. But if the multicollinearity is less perfect, it might dilute the effects of a variable and inappropriately lead us to confirm the null hypothesis (since error bars get really big when multicollinearity is in play, and they get bigger the more multicollinear our findings are). An actual relationship will suddenly appear spuriously to be spurious.

0.3 Problem 3

Using the turnout data:

```r
# Calculate leverage (diagonal of hat matrix)
turnout = read.csv("/Users/gideon/Desktop/Desktop/Datasets/turnout.csv")
turnout_clean = drop_na(turnout)

model <- lm(Turnout ~ GDP + Temperature, data = turnout_clean)

#turnout_matrix =
# data.frame(Year = turnout_clean$Year,
#            Temperature = turnout_clean$Temperature,
#            GDP = turnout_clean$GDP)

BigX = model.matrix(model)

# Method 1: Using hatvalues()
leverage1 <- hatvalues(model)

# Method 2: Calculate manually
hat_matrix <- BigX %*% solve(t(BigX) %*% BigX, tol = 0) %*% t(BigX)
leverage2 <- diag(hat_matrix)

# Compare
head(leverage1)
```

```
         1          2          3          4          5          6
0.03624436 0.12541190 0.03550707 0.07860531 0.03497553 0.03715984
```

```r
head(leverage2)
```

```
         1          2          3          4          5          6
```

```
0.03624436 0.12541190 0.03550707 0.07860531 0.03497553 0.03715984
```

```
tail(leverage1)
```

```
       33         34         35         36         37         38
0.0854872 0.1535744 0.1679493 0.1607587 0.1934831 0.3773473
```

```
leverage_data = data.frame(leverage = leverage1)
leverage_data %>% top_n(9)
```

```
Selecting by leverage
```

```
    leverage
2  0.1254119
7  0.1328942
15 0.1559861
23 0.1551246
34 0.1535744
35 0.1679493
36 0.1607587
37 0.1934831
38 0.3773473
```

It looks like the top five were 15, 35, 36, 37, and 38.

Note that my initial try kept doing weird things so I tried to grab a matrix differently using `model.matrix`, relying in part on the answer given by Google's AI when I searched for information on the errors I was getting. This worked.

Questions: 1. What does leverage measure? Why do we care about high leverage points?

Leverage basically measures how much a certain observation influences the estimates for the data as a whole, which we can tell by seeing how much the coefficients change when that

observation is excluded (the DFBETA) or in some other ways.

2. What is the average leverage value? What's the theoretical value?

The mean leverage value for this data is 0.079.

```
mean(leverage1)
```

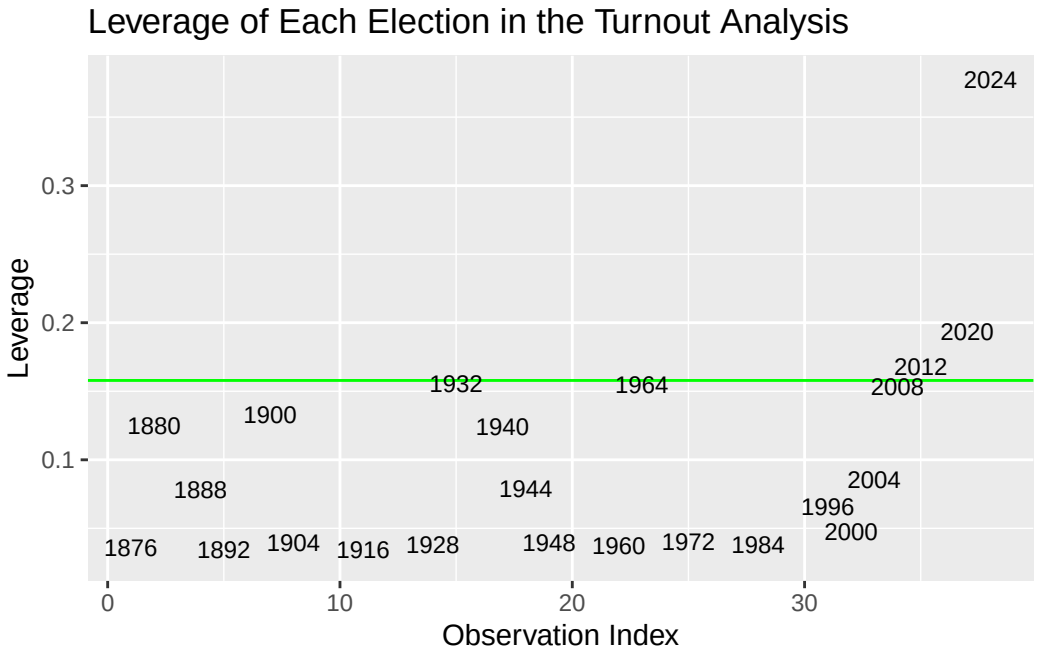
```
[1] 0.07894737
```

I'm not sure what theoretical values of leverage are — it seems like leverage can only exist for actual data?

3. Create a plot of leverage vs. observation index. Add a horizontal line at $2p/n$ (where p = number of parameters).

```
two_p_over_n = 6/nrow(turnout_clean)
leverage_data = mutate(leverage_data, Year = turnout_clean$Year)


ggplot(aes(y = leverage, x = 1:nrow(leverage_data)), data = leverage_data) +
  labs(title = "Leverage of Each Election in the Turnout Analysis", x = "Observation Ind
  geom_hline(yintercept = two_p_over_n, color = "green") +
  geom_text(aes(label = Year), size = 3, check_overlap = TRUE)
```



0.4 Problem 4

4a. 1. Define DFBETA and Cook's Distance. What does each measure? The DFBETA for an observation is relative to a particular coefficient or coefficients; it measures how much a coefficient changes when that observation is removed. It's defined by $_{(i)}$, or $(XX)^1\mathbf{X}_i e_i$. Cook's Distance is a measure of "DFFIT", difference in fit, and measures the effect of an observation on all the variables at once, including the intercept. Cook's Distance is defined as $D_i = [(n\ \ k\ \ 1)/(k+1)]\ \text{Œ}\ (h_{ii}e_i^2)/\mathbf{ee}$.

2. What's the difference between an outlier and an influential point? An outlier is far from the rest of the data, with a Y value pretty far from the Y-value predicted given its X-value, but might not play much of a role in the outcome. If an outlier is right over or right under the center of mass of the distribution, it won't shift the coefficients much in one direction or another.

4b. Using the turnout model:

```r
# Calculate DFBETA
dfbeta_frame = as.data.frame(dfbeta(model))
dfbeta_temperature = data.frame(Year = leverage_data$Year, DFBETA = dfbeta_frame$Tempera
# Plot DFBETA for Temperature coefficient
ggplot(aes(y = DFBETA, x = 1:nrow(dfbeta_temperature)), data = dfbeta_temperature) +
  labs(title = "DFBETAs of Temperature in Each Election", x = "Observation Index", y = "
  geom_text(aes(label = Year), size = 3, check_overlap = TRUE)
```

Questions: 1. Which observations are influential for the Temperature coefficient? The big years are 1880, 1888, 1900, and 1932. 1944 and 1996 are marginal.

2. What happens to the Temperature coefficient if you remove the most influential observation?

```r
turnout_clean_temppurge = turnout_clean[-c(2,4,7,15), ]
model_temppurge <- lm(Turnout ~ GDP + Temperature, data = turnout_clean_temppurge)
summary(model)
```

```
Call:
lm(formula = Turnout ~ GDP + Temperature, data = turnout_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-0.13323 -0.05916 -0.01721  0.02748  0.19650

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.751e-01  5.458e-01   1.054    0.299
```

```
GDP         -3.525e-09  3.006e-09  -1.173    0.249
Temperature  9.409e-04  1.066e-02   0.088    0.930

Residual standard error: 0.09201 on 35 degrees of freedom
Multiple R-squared:  0.06614,   Adjusted R-squared:  0.01278
F-statistic: 1.239 on 2 and 35 DF,  p-value: 0.3019
```

```
summary(model_temppurge)
```

```
Call:
lm(formula = Turnout ~ GDP + Temperature, data = turnout_clean_temppurge)

Residuals:
      Min        1Q    Median        3Q       Max
-0.118556 -0.044840 -0.007916  0.028759  0.212812

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.819e-01  6.125e-01   1.603    0.119
GDP         -1.034e-09  3.058e-09  -0.338    0.738
Temperature -7.277e-03  1.195e-02  -0.609    0.547

Residual standard error: 0.08293 on 31 degrees of freedom
Multiple R-squared:  0.05976,   Adjusted R-squared:  -0.0008966
F-statistic: 0.9852 on 2 and 31 DF,  p-value: 0.3847
```

Oddly, with those observations removed, I get a larger temperature coefficient. But also, it doesn't seem to be significant anyhow.

3. Calculate Cook's Distance for all observations. Which observations have Cook's D > 0.5?

```
cooksd_model = data.frame(Year = turnout_clean$Year, Cooks_Distance = cooks.distance(mod
big_cooks = filter(cooksd_model, Cooks_Distance > 0.05)
```

Cook's Distance for all the observations is less than 0.5. Only seven observations even have Cook's D > *0.05*; these are 1876, 1880, 1888, 1900, 1932, 2020, and 2024.

4c. Case Study Analysis: Identify the most influential observation in your model and analyze it:

```
# Find most influential observation
Biggest_Cooks = cooksd_model$Year[which.max(cooksd_model$Cooks_Distance)]


# Analyze this observation
leverage_data$leverage[2]
```

```
[1] 0.1254119
```

```
residuals(model)[2]
```

```
        2
0.1692336
```

```
# Run model without this observation
turnout_clean_infpurge = turnout_clean[-c(2), ]
model_infpurge <- lm(Turnout ~ GDP + Temperature, data = turnout_clean_infpurge)


# Compare coefficients
coef(model_infpurge)
```

```
  (Intercept)           GDP   Temperature
```

```
 2.233047e-01 -4.454949e-09  7.679860e-03
```

```
coef(model)
```

```
  (Intercept)           GDP   Temperature
 5.750768e-01 -3.524858e-09  9.408781e-04
```

```
summary(model_infpurge)

summary(model)
```

I did a closer look to see what the p-values are; generally not high in either case.

Questions: 1. Why is this observation influential? Consider its leverage and residual.

The election of 1880 (second observation) has a leverage of ~0.125, and a residual of 0.169. It's not one of the highest-leverage observations (it's the ninth-highest leverage observation, specifically), and its residual is relatively high but not the highest. But between those two factors, it seems pretty important, with a Cook's Distance of 0.184.

2. Should this observation be removed? What are the ethical and methodological considerations?

So, 1880 was a weird year. Presidential elections were direct for the first time, so it may be unusual in that regard. It also had a weird winter. Removing it decreases the effect of temperature by a full order of magnitude, reducing its effects to almost nothing. I think this may be appropriate, though, because a freak weather event coincided with a freak political event, which magnifies the apparent effects.

I would suggest removing this observation, though our n is already pretty tiny. If I were doing this for real, I would worry a bit more, but in either case this is a catastrophically high p-value.

3. What would you recommend to a researcher who found such an influential observation in their data?

I would suggest that someone should remove it, but explain at length why that particular case is *so* exceptional that it does not have a similar causal process to other cases.