Linear Regression

Adding variables to the model and using transformation

Jenine Harris
Brown School

Importing and merging data sources

```
# distance to syringe program data
dist.ssp <- read.csv(file = "/Users/harrisj/Box/teaching/Teaching/Fall20

# regression
dist.by.unins <- lm(formula = dist_SSP ~ pctunins,
                    data = dist.ssp, na.action = na.exclude)
summary(dist.by.unins)
```

```
##
## Call:
## lm(formula = dist_SSP ~ pctunins, data = dist.ssp, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -217.71  -60.86  -21.61   47.73  290.77
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.4798    10.1757   1.226    0.221
## pctunins      7.8190     0.7734  10.110   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85.91 on 498 degrees of freedom
```

Codebook

Leslie looked through the variables and the codebook and determined that the variables had the following meanings:

- county: the county name
- STATEABBREVIATION: the two-letter abbreviation for the state the county is in
- dist_SSP: distance in miles to the nearest syringe services program
- HIVprevalence: people age 13 and older living with diagnosed HIV per 100,000
- opioid_RxRate: number of opioid prescriptions per 100 people
- pctunins: percentage of the civilian noninstitutionalized population with no health insurance coverage
- metro: county is non-metro, which includes open countryside, rural towns, or smaller cities with up to 49,999 people, or metro

Adding a binary variable to the model

- Uninsured percentage accounted for 17% of the variation in distance to syringe program for counties, leaving about 83% still unexplained.
- Add in more variables to see if they account for some of the variation in distance to syringe program.
- Perhaps bigger cities are more likely to have these programs, so the `metro` variable seems like it might help to explain how far away a county is from a syringe program.

```
# linear regression distance to syringe program by
# uninsured percent and metro status in 500 counties
dist.by.unins.metro <- lm(formula = dist_SSP ~ pctunins +
                          metro, data = dist.ssp)
summary(dist.by.unins.metro)
```

```
##
## Call:
## lm(formula = dist_SSP ~ pctunins + metro, data = dist.ssp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -219.80  -60.07  -18.76   48.33  283.96
##
```

Interpreting the multiple regression model results

- The results who low p-values on the rows for `pctunins` and `metro`, which indicated that percentage uninsured and metro status both statistically significantly help to explain the distance to a syringe program.
- The model was statistically significant, with an F-statistic of F(2, 497) = 58.88 and a p-value of < .001.
- The R^2_{adj} = .1883 indicated that 18.83% of the variation in distance to syringe program was accounted for by this model that has uninsured percentage and metro status in it.
- This is higher than the R^2_{adj} from the simple linear model with just uninsured in it. F
- inally, the coefficient for `pctunins` was 7.30, so for every 1% more uninsured in a county, the distance to a syringe program is 7.30 miles further.

Interpreting the binary predictor

- The row with `metrononmetro` is confusing!
- This row is showing the variable name `metro` and the name of the category within the `metro` variable.
- In this case, the coefficient refers to the `nonmetro` counties and the `metro` counties are the reference group.
- The non-metro counties are 28.0525 miles further away from the nearest syringe program than the metro counties.

Visualize a model with a binary predictor

- What happens with a binary variable, is that the slope of the line does not change for each group, but the y-intercept changes.

```
# graphing the regression model with percent uninsured and metro
dist.ssp %>%
  ggplot(aes(x = pctunins, y = dist_SSP, group = metro)) +
  geom_line(data = broom::augment(dist.by.unins.metro),
            aes(y = .fitted, linetype = metro)) +
  geom_point(aes(color = metro), alpha = .4, size = 2) +
  theme_minimal() +
  scale_color_manual(values = c("dodgerblue2", "deeppink"), name = "Coun
  labs(x = "Miles to nearest syringe program",
       y = "County percent uninsured",
       title = "Miles to nearest syringe program based on uninsured perc
  scale_linetype_manual(values = c(1,2), name = "Regression line\n(predi
```

Visualize a model with a binary predictor

- What happens with a binary variable, is that the slope of the line does not change for each group, but the y-intercept changes.

Using the multiple regression model

Reviewing the regression model can also help:

- distance to syringe program = 3.42 + 7.3 * percent uninsured + 28.05 * non-metro

Substitute in values for an example county with 10% uninsured in a **non**-metro area:

- distance to syringe program = 3.42 + 7.3 * 10 + 28.05 * 1
- distance to syringe program = 104.48

Substituted in the values for an example county with 10% uninsured in a **metro** area:

- distance to syringe program = 3.42 + 7.3 * 10 + 28.05 * 0
- distance to syringe program = 76.43
- A county with 10% uninsured in a metro area would have to travel 28.05 fewer miles to a syringe program given the coefficient for `metro` in the model.
- Notice that the two lines in the figure look about 28 miles apart, which is consistent with the interpretation of the `metro` coefficient.

Adding more variables to the model

- While normally distributed predictors is not an assumption, transforming variables in a model is one strategy used at times in order to meet the other assumptions for linear regression (or for other reasons).
- A review of the HIV prevalence variable finds it to be non-normal but the log transformation of HIV prevalence looks the most normally distributed of the transformation options (see Chapter 8 for a reminder).
- The cube root of `dist_SSP` looks most normal for the outcome.
- Add the transformations directly into the `lm()` function code for the larger model.

```
# linear regression of distance by percent uninsured, HIV prevalence,
# metro status
dist.full.model <- lm(formula = (dist_SSP)^(1/3) ~ pctunins +
                        log(x = HIVprevalence) + metro,
                      data = dist.ssp,
                      na.action = na.exclude)
summary(dist.full.model)
```

```
##
## Call:
## lm(formula = (dist_SSP)^(1/3) ~ pctunins + log(x = HIVprevalence) +
```

Interpret the model results

- The model was statistically significant, with an F-statistic of F(3, 426) = 44.16 and a p-value of < .001.
- The R^2_{adj} = .2318 indicated that 23.18% of the variation in distance to syringe program is accounted for by this model that has HIV prevalence, uninsured percentage, and metro status in it.
- This is higher than the R^2_{adj} from the previous two models.
- The coefficient for `pctunins` was .1127, so for every 1% more uninsured in a county, the *cube root* of the distance to a syringe program is expected to change by .1127.
- The positive and significant coefficient of .48808 for `metronon-metro` in the output suggests that non-metro areas are further from syringe programs.
- The log of HIV prevalence was not statistically significantly associated with distance to syringe program.
- Note that the denominator degrees of freedom value is now 426, which is a lot lower than the 498 from the simple linear regression model they first estimated.
    - The log value of zero is undefined, so they probably lost some counties with 0 HIV prevalence when they transformed this variable using the `log()`.

No multicollinearity assumption for multiple regression

- There is one additional assumption to be checked when there are multiple *continuous* predictor variables in a model.
- There are two continuous predictors in `dist.full.model`, so check this additional assumption.
- In addition to the assumptions checked with the earlier model, when a model has more than one continuous predictor, there is an assumption of **no perfect multicollinearity**.
- Multicollinearity is when two variables are highly correlated and therefore are very similar to one another.
- When two variables are similar to one another, they are both bringing the same information into the regression model.
- This redundancy can be a problem for model estimation, so variables that are too similar should not be in a model together.

Using correlation to check multicollinearity

- There are several ways to check for multicollinearity.
- The first is to examine correlations between any continuous variables in a model before estimating the model.
- In this case, `pctunins` and the transformed `log(x = HIVprevalence)` are continuous.
- The correlation between these can be computed using the `cor()` function.

```
# correlations among continuous variables in the full model
dist.ssp %>%
  mutate(log.HIVprev = log(x = HIVprevalence)) %>%
  drop_na(log.HIVprev) %>%
  summarize(cor.hiv.unins = cor(x = log.HIVprev, y = pctunins))
```

```
##   cor.hiv.unins
## 1     0.2444709
```

- The result was a weak correlation of .24 between percent uninsured and the transformed value of `HIVprevalence`.
- If the absolute value of the correlation coefficient is .7 or higher, this would indicate a strong relationship with a large amount of shared variance between the two variables and therefore a

Using variance inflation factors (VIF) to check multicollinearity

- The other way to identify problems with multicollinearity is through the use of **Variance Inflation Factor** or **VIF** statistics.
- The VIF statistics are calculated by running a separate regression model for each of the predictors where the predictor is the outcome and everything else stays in the model as a predictor.
- With this model, for example, the VIF for the `pctunins` variable would be computed:
  - `pctunins = log(HIVprevalence) + metro`
- The R^2 would be used to determine the VIF by substituting it into: $VIF_{pctunins} = \frac{1}{1-R^2}$
- The result will be `1` if there is no shared variance at all.
- If there is any shared variance, the VIF will be greater than one.
- If the VIF is large, this indicates that `pctunins` shares a lot of variance with the `metro` and `log(HIVprevalence)` variables.
- A VIF of 2.5, for example, would indicate that the R^2 was .60 and so 60% of the variation in `pctunins` was explained by `metro` and `log(HIVprevalence)`.

Computing VIF in R

- Use `vif()` command to check VIF values for the model above:

```
# VIF for model with poverty
car::vif(dist.full.model)
```

```
##              pctunins log(x = HIVprevalence)                    metro
##              1.165165                1.207491                 1.186400
```

- The VIF values are small, especially given that the lower limit of the VIF is one.
- This confirmed no problem with multicollinearity with this model.
- The model meets the assumption of no perfect multicollinearity.
- Kiara explained to Leslie that the rest of the assumption checking and diagnostics are conducted and interpreted in the same way as they were for the simple linear regression model.
- Leslie was interested in checking some of the other assumptions since they now have transformed variables in the model.

Checking linearity for multiple regression

- Check each of the continuous predictors for a linear relationship with the outcome, which is now the cube root of `dist_SSP`.

```
# log of HIV prevalence and cube root of distance to needle exchange
dist.ssp %>%
  ggplot(aes(x = log(HIVprevalence), y = (dist_SSP)^(1/3))) +
  geom_point(aes(size = "County"), color = "#7463AC", alpha = .6) +
  geom_smooth(aes(color = "Linear fit line"), method = "lm", se = FALSE)
  geom_smooth(aes(color = "Loess curve"), se = FALSE) +
  theme_minimal() +
  labs(y = "Cube root of miles to syringe program",
       x = "Log of HIV prevalence",
       title = "Relationship between the log of HIV prevalence and trans
  scale_color_manual(values = c("gray60", "deeppink"), name = "") +
  scale_size_manual(values = 2, name = "")
```

Checking linearity

```
# percent uninsured and cube root of distance to needle exchange
dist.ssp %>%
  ggplot(aes(x = pctunins, y = (dist_SSP)^(1/3))) +
  geom_point(aes(size = "County"), color = "#7463AC", alpha = .6) +
  geom_smooth(aes(color = "Linear fit line"), method = "lm", se = FALSE)
  geom_smooth(aes(color = "Loess curve"), se = FALSE) +
  theme_minimal() +
  labs(y = "Cube root of miles to syringe program",
       x = "Percent uninsured",
       title = "Relationship between percentage uninsured and transforme
  scale_color_manual(values = c("gray60", "deeppink"), name = "") +
  scale_size_manual(values = 2, name = "")
```

Checking the homoscedasticity assumption for multiple regression

- The Breusch-Pagan can be used to test the null hypothesis that the variance is constant:

```
# testing for equal variance
const.var.test.full <- lmtest::bptest(formula = dist.full.model)
const.var.test.full
```

```
##
##     studentized Breusch-Pagan test
##
## data:  dist.full.model
## BP = 36.288, df = 3, p-value = 6.51e-08
```

- The Breusch-Pagan test statistic has a tiny p-value associated with it (BP = 36.29; p <.001), indicating that the null hypothesis would be rejected.
- The assumption of constant variance is not met.

Testing the independence of residuals assumption

- The Durbin-Watson tests the null hypothesis that the residuals are independent.

```
# test independence of residuals
lmtest::dwtest(formula = dist.full.model)
```

```
##
##     Durbin-Watson test
##
## data:  dist.full.model
## DW = 1.9631, p-value = 0.3494
## alternative hypothesis: true autocorrelation is greater than 0
```

- The D-W statistic was near 2 and the p-value was high, so Leslie concluded that the null hypothesis was retained.
- Since the null hypothesis was that the residuals are independent, this assumption was met.

Testing the normality of residuals assumption

- The last assumption to check is normality of residuals.

```
# check residual plot of uninsured percent and distance to syringe progr
data.frame(dist.full.model$residuals) %>%
  ggplot(aes(x = dist.full.model.residuals)) +
  geom_histogram(fill = "#7463AC", col = "white") +
  theme_minimal() +
  labs(x = "Residual (difference between observed and predicted values)"
       y = "Number of counties",
       title = "Distribution of residuals for model explaining distance
```

Using the Partial-F test to choose a model

- There are a few things to think about in selecting a model before thinking about how it performed.
- First, the model should address the research question of interest.
- Second, the model should include variables---if any---that have been demonstrated important in the past to help explain the outcome.
- For example, a statistical model explaining lung cancer should include smoking status since it has been demonstrated by many studies to have a strong relationship to lung cancer.
- After answering the research question and including available variables demonstrated important in the past, choosing a model can still be complicated.
- One tool for choosing between two linear regression models is a statistical test called the Partial-F test.
- The Partial-F test compares the fit of two **nested** models to determine if the additional variables in the larger model improved the model fit enough to warrant keeping the variables and interpreting the more complex model.

Conducting the partial-F test manually

- The Partial-F test can be conducted by hand using the Partial-F equation.

$$F_{partial} = \frac{\frac{R^2_{full} - R^2_{reduced}}{q}}{\frac{1 - R^2_{full}}{n-p}}$$

Where:

- R^2_{full} is the R^2 for the larger model
- $R^2_{reduced}$ is the R^2 for the smaller nested model
- n is the sample size
- q is difference in the number of parameters for the two models
- p is the number of parameters in the larger model

Compute partial-F

The $F_{partial}$ statistic has q and n - p degrees of freedom. To compare the `dist.by.unins` model with the `dist.by.unins.metro` model, substitute their values into the equation and compute as in Equation \@ref(eq:partialf2).

```
##
## Call:
## lm(formula = dist_SSP ~ pctunins, data = dist.ssp, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -217.71  -60.86  -21.61   47.73  290.77
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.4798    10.1757   1.226    0.221
## pctunins      7.8190     0.7734  10.110   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 85.91 on 498 degrees of freedom
## Multiple R-squared:  0.1703,    Adjusted R-squared:  0.1686
## F-statistic: 102.2 on 1 and 498 DF,  p-value: < 2.2e-16

##
## Call:
## lm(formula = dist_SSP ~ pctunins + metro, data = dist.ssp)
##
```

Use R to conduct the partial-F test

- Use the `anova()` function.
- Enter the name of the smaller model first and then the larger model into `anova()` and the function will compare the two models using a Partial-F test.

```
# partial F test for dist.by.unins and dist.by.unins.metro
anova(object = dist.by.unins, dist.by.unins.metro)
```

```
## Analysis of Variance Table
##
## Model 1: dist_SSP ~ pctunins
## Model 2: dist_SSP ~ pctunins + metro
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    498 3675855
## 2    497 3581712  1     94143 13.063 0.0003318 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Partial-F was 13.063 and the p-value was .0003.

NHST Step 1: Write the null and alternate hypotheses

H0: The larger model is no better than the smaller model at explaining the outcome

HA: The larger model is better than the smaller model at explaining the outcome

NHST Step 2: Compute the test statistic

The Partial-F is 13.063 with 1 and 497 degrees of freedom.

NHST Step 3: Calculate the probability that your test statistic is at least as big as it is if there is no relationship (i.e., the null is true)

The p-value is very small (p = .0003), so the probability is tiny that the test statistic would be this big or bigger if the null hypothesis were true.

NHST Step 4 & 5: Reject or retain the null hypothesis based on the probability

- The null hypothesis is rejected; it is unlikely that the null hypothesis is true.
- This suggests that the model with uninsured percentage and metro status was a better model for reporting than the simple linear model.

Write the final interpretation

- (be sure to test assumptions for your final model before reporting!)

```
##
## Call:
## lm(formula = dist_SSP ~ pctunins + metro, data = dist.ssp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -219.80  -60.07  -18.76   48.33  283.96
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.4240    10.3621   0.330 0.741212
## pctunins         7.3005     0.7775   9.389  < 2e-16 ***
## metronon-metro  28.0525     7.7615   3.614 0.000332 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 84.89 on 497 degrees of freedom
## Multiple R-squared:  0.1915,    Adjusted R-squared:  0.1883
## F-statistic: 58.88 on 2 and 497 DF,  p-value: < 2.2e-16
```

```
##                     2.5 %    97.5 %
## (Intercept)    -16.934973 23.782947
## pctunins         5.772859  8.828114
## metronon-metro  12.803152 43.301754
```