Problem set 7: Worked solutions

Statistics and statistical programming
Northwestern University
MTS 525

Aaron Shaw

November 16, 2020

Contents

Programming challenges

I'll start by loading some useful libraries...

```
library(tidyverse)
library(ggfortify)
```

PC1. Import and update

Following the link and instructions in the problem set itself:

```
hibbs <- read.table(url("https://github.com/avehtari/ROS-Examples/raw/master/ElectionsEconomy/data/hibbs

hibbs
```

```
##    year growth  vote inc_party_candidate other_candidate
## 1  1952   2.40 44.60           Stevenson      Eisenhower
## 2  1956   2.89 57.76          Eisenhower       Stevenson
## 3  1960   0.85 49.91               Nixon         Kennedy
## 4  1964   4.21 61.34             Johnson       Goldwater
## 5  1968   3.02 49.60            Humphrey           Nixon
```

```
## 6  1972   3.62 61.79              Nixon        McGovern
## 7  1976   1.08 48.95               Ford          Carter
## 8  1980  -0.39 44.70             Carter          Reagan
## 9  1984   3.86 59.17             Reagan         Mondale
## 10 1988   2.27 53.94          Bush, Sr.         Dukakis
## 11 1992   0.38 46.55          Bush, Sr.         Clinton
## 12 1996   1.04 54.74            Clinton            Dole
## 13 2000   2.36 50.27              Gore       Bush, Jr.
## 14 2004   1.72 51.24          Bush, Jr.           Kerry
## 15 2008   0.10 46.32             McCain           Obama
## 16 2012   0.95 52.00              Obama          Romney
```

And here I'll rbind up that dataset with a new row for 2016. Note that by

```
newrow <- list(2016, 2, 51.1, "Clinton", "Trump")
hibbs <- rbind(hibbs, newrow)

hibbs
```

```
##    year growth  vote inc_party_candidate other_candidate
## 1  1952   2.40 44.60           Stevenson      Eisenhower
## 2  1956   2.89 57.76          Eisenhower       Stevenson
## 3  1960   0.85 49.91               Nixon         Kennedy
## 4  1964   4.21 61.34             Johnson       Goldwater
## 5  1968   3.02 49.60            Humphrey           Nixon
## 6  1972   3.62 61.79              Nixon        McGovern
## 7  1976   1.08 48.95               Ford          Carter
## 8  1980  -0.39 44.70             Carter          Reagan
## 9  1984   3.86 59.17             Reagan         Mondale
## 10 1988   2.27 53.94          Bush, Sr.         Dukakis
## 11 1992   0.38 46.55          Bush, Sr.         Clinton
## 12 1996   1.04 54.74            Clinton            Dole
## 13 2000   2.36 50.27              Gore       Bush, Jr.
## 14 2004   1.72 51.24          Bush, Jr.           Kerry
## 15 2008   0.10 46.32             McCain           Obama
## 16 2012   0.95 52.00              Obama          Romney
## 17 2016   2.00 51.10            Clinton           Trump
```

PC2. Summarize and visualize the data

I'll start with some basic summary info about the two variables we care about most as well as univariate boxplots and a bivariate scatterplot.

```
summary(hibbs$growth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -0.390   0.950   2.000   1.904   2.890   4.210
```

```
sd(hibbs$growth)
```

```
## [1] 1.351453
```

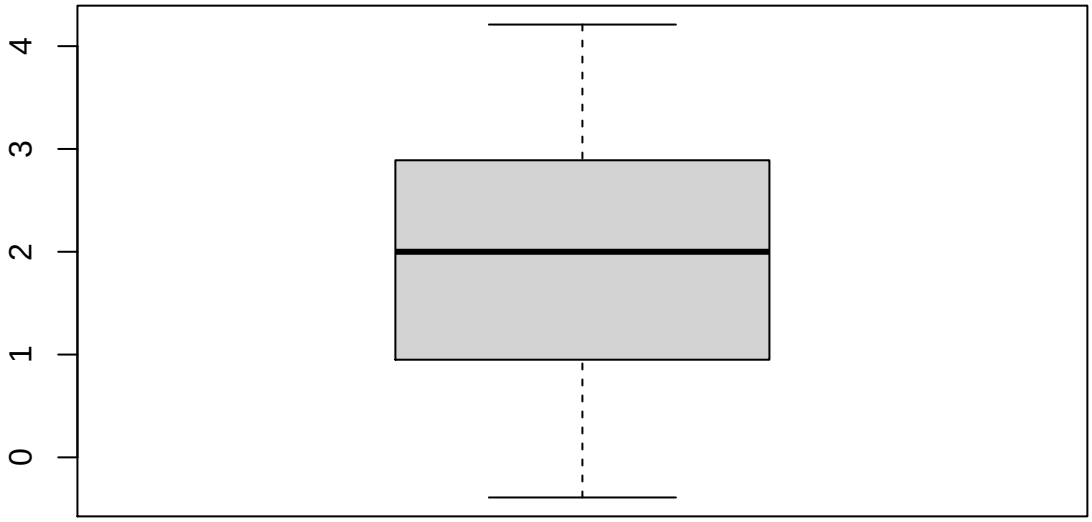
```
summary(hibbs$vote)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   44.60   48.95   51.10   52.00   54.74   61.79
```

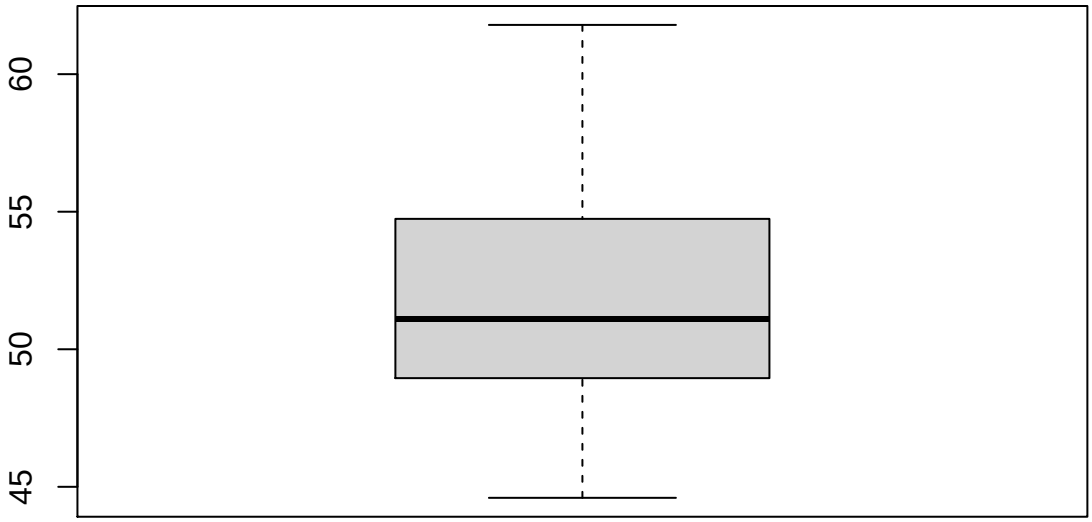
```r
sd(hibbs$vote)
```

```
## [1] 5.435781
```

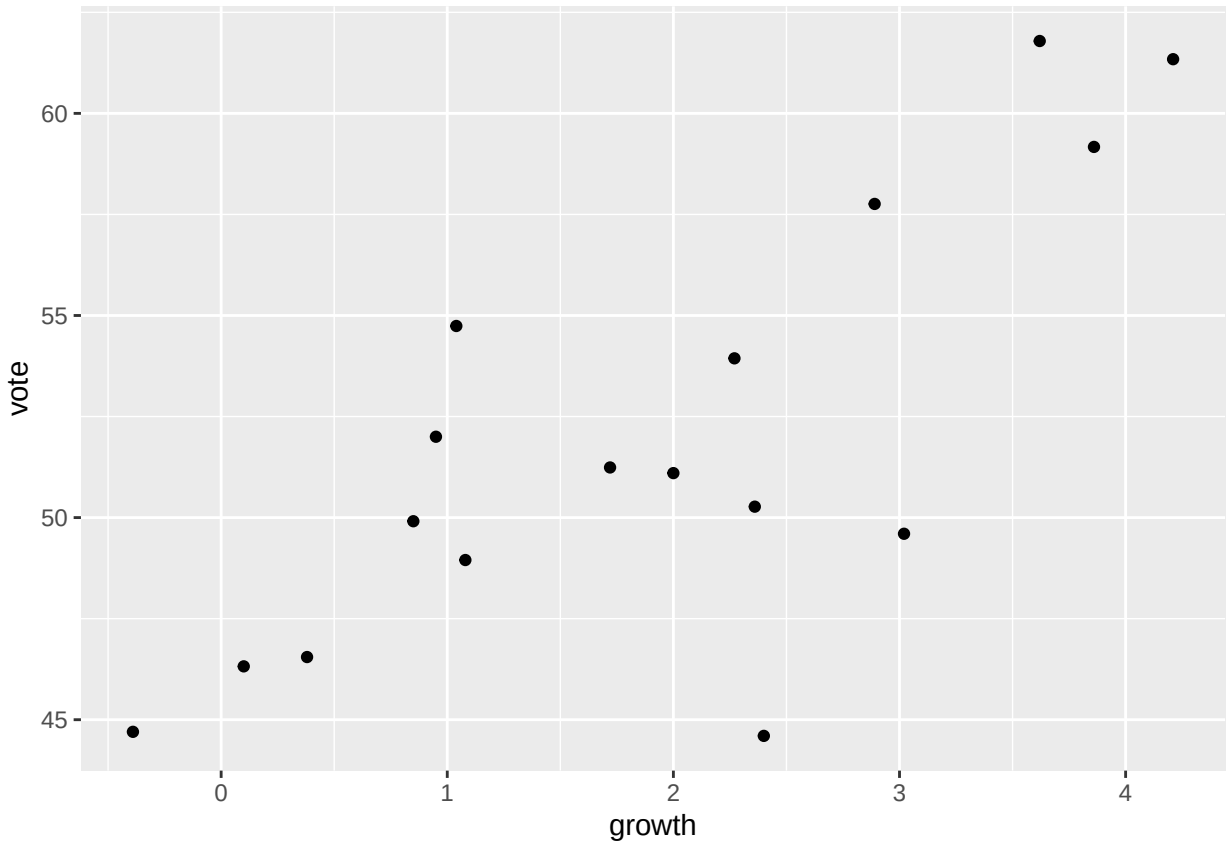
```r
boxplot(hibbs$growth)
```




```r
boxplot(hibbs$vote)
```




```r
p <- ggplot(data = hibbs, aes(x = growth, y = vote, label = year)) +
  geom_point()
p
```



Just to supplement that, I'm going to replace the points with labels for the years.

```
p + geom_label()
```

It's worth noting that this data looks like a good candidate for linear model. Both variables appear to follow quite normal distributions and there's a seemingly clear, positive, linear trend when we plot them against each other.

PC3. Covariance and correlation

```
with(hibbs, cov(growth, vote))
```

```
## [1] 5.582173
```

```
with(hibbs, cor(growth, vote))
```

```
## [1] 0.7598722
```

Note that since this is a bivariate analysis, you could also learn quite a bit by calculating a more formal hypothesis test and confidence intervals around the correlation. Here's an example of that. Please see the documentation for `cor.test()` for more.

```
with(hibbs, cor.test(growth, vote))
```

```
##
##  Pearson's product-moment correlation
##
## data:  growth and vote
## t = 4.5271, df = 15, p-value = 0.000401
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4398866 0.9086515
## sample estimates:
```

```
##       cor
## 0.7598722
```

PC4. Fit and summarize a linear model

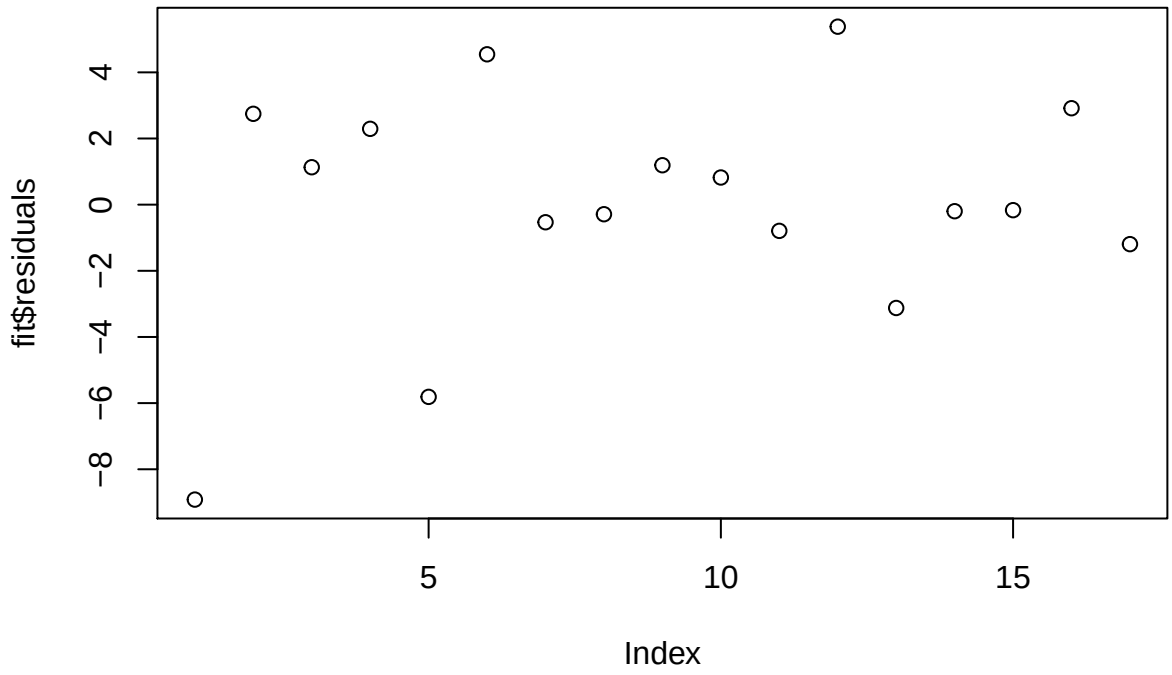
```
fit <- lm(vote ~ growth, data = hibbs)
summary(fit)
```

```
##
## Call:
## lm(formula = vote ~ growth, data = hibbs)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9162 -0.7924 -0.1666  2.2918  5.3804
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  46.1810     1.5604  29.595 1.02e-14 ***
## growth        3.0563     0.6751   4.527 0.000401 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.65 on 15 degrees of freedom
## Multiple R-squared:  0.5774, Adjusted R-squared:  0.5492
## F-statistic:  20.5 on 1 and 15 DF,  p-value: 0.000401
```

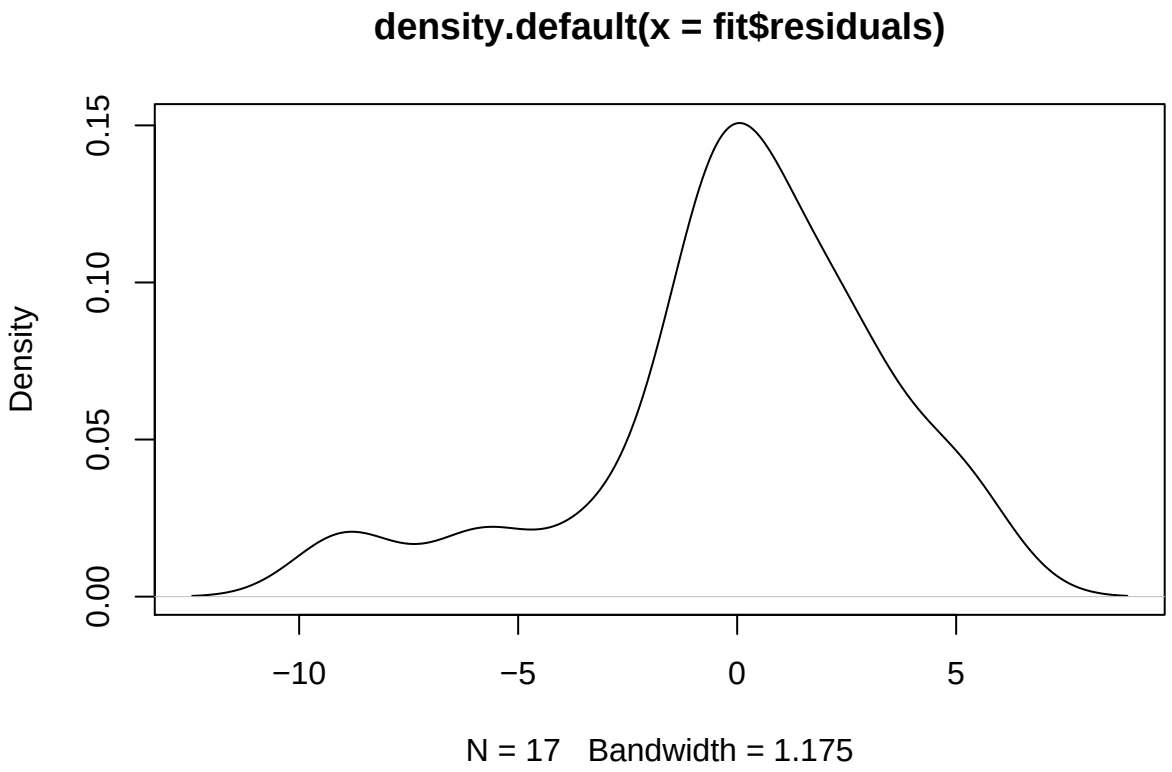
PC5 Assess model fit

For assessing the model fit and the conditions necessary to identify an OLS model, we need a few things. The first of these are actually recovered from our summary statistics and scatterplot above: the variables both appear normally distributed and they appear to possess a clear linear trend when plotted against each other. Onwards to the stuff the problem asks us to visualize.

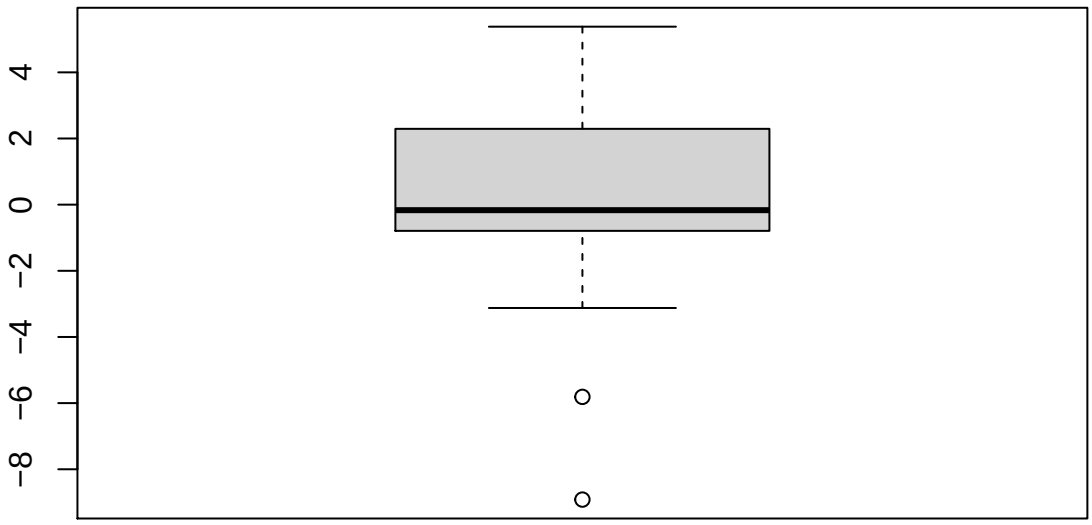
```
## These first three just look at the residuals alone
plot(fit$residuals)
```



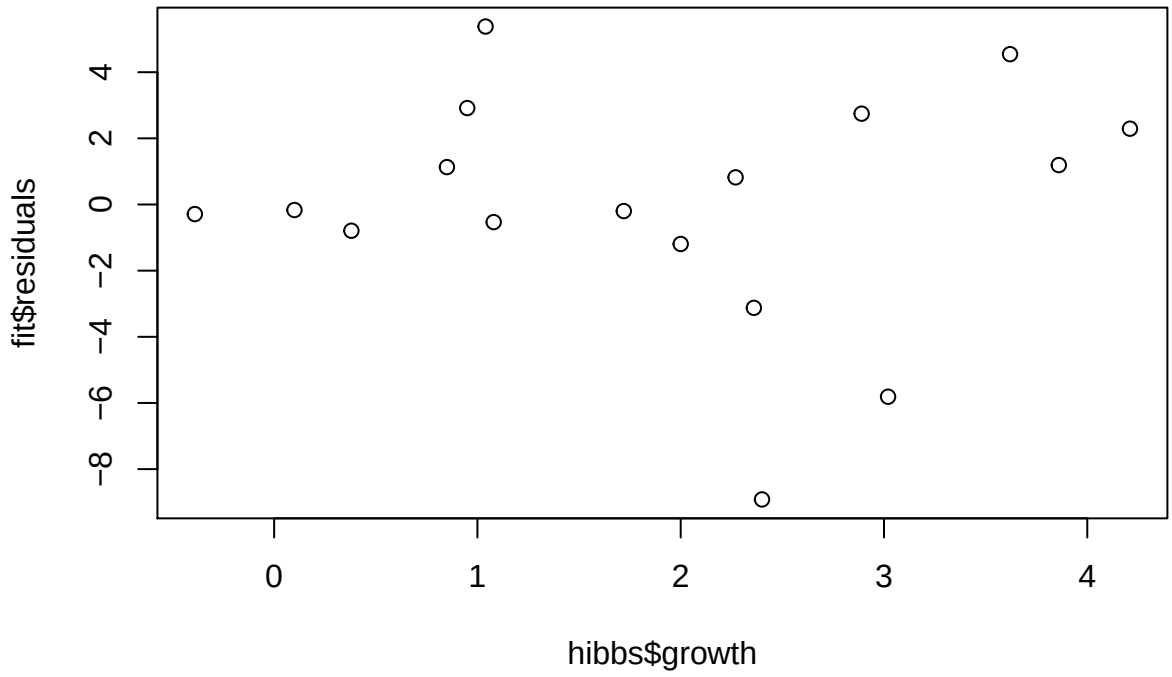

```
plot(density(fit$residuals))
```



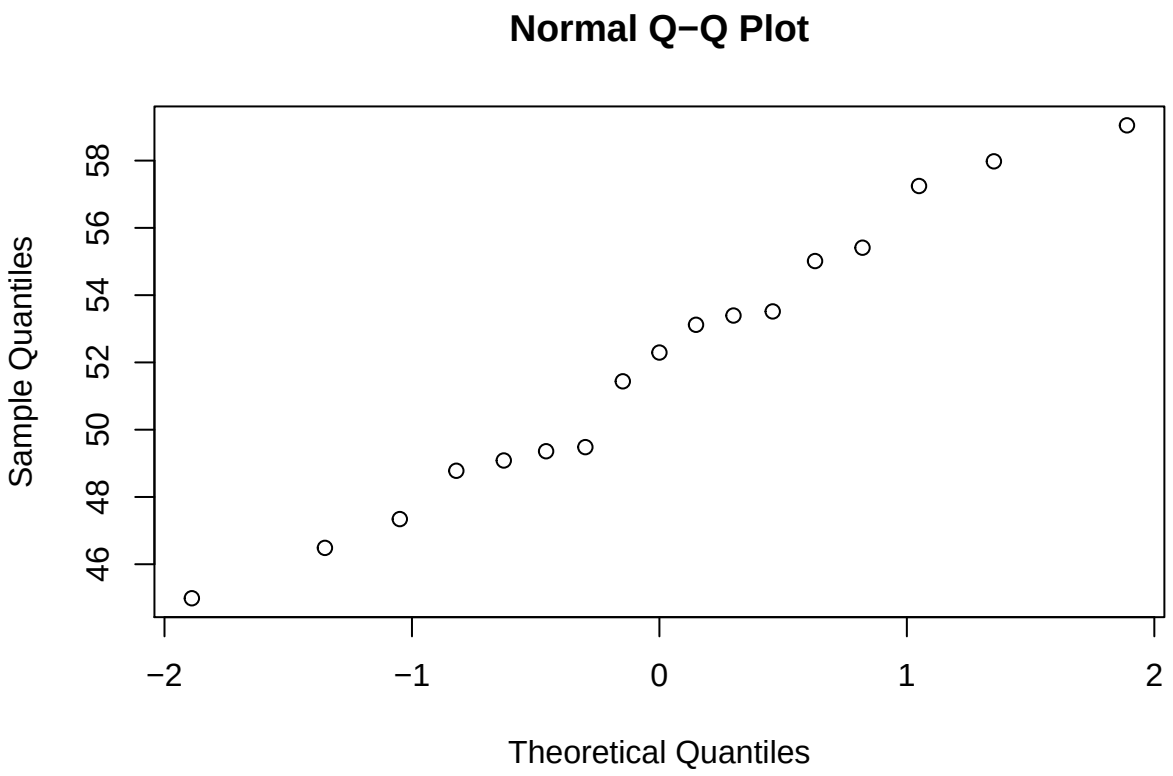

```
boxplot(fit$residuals)
```




```
## This considers residuals against the values of X:
plot(hibbs$growth, fit$residuals)
```



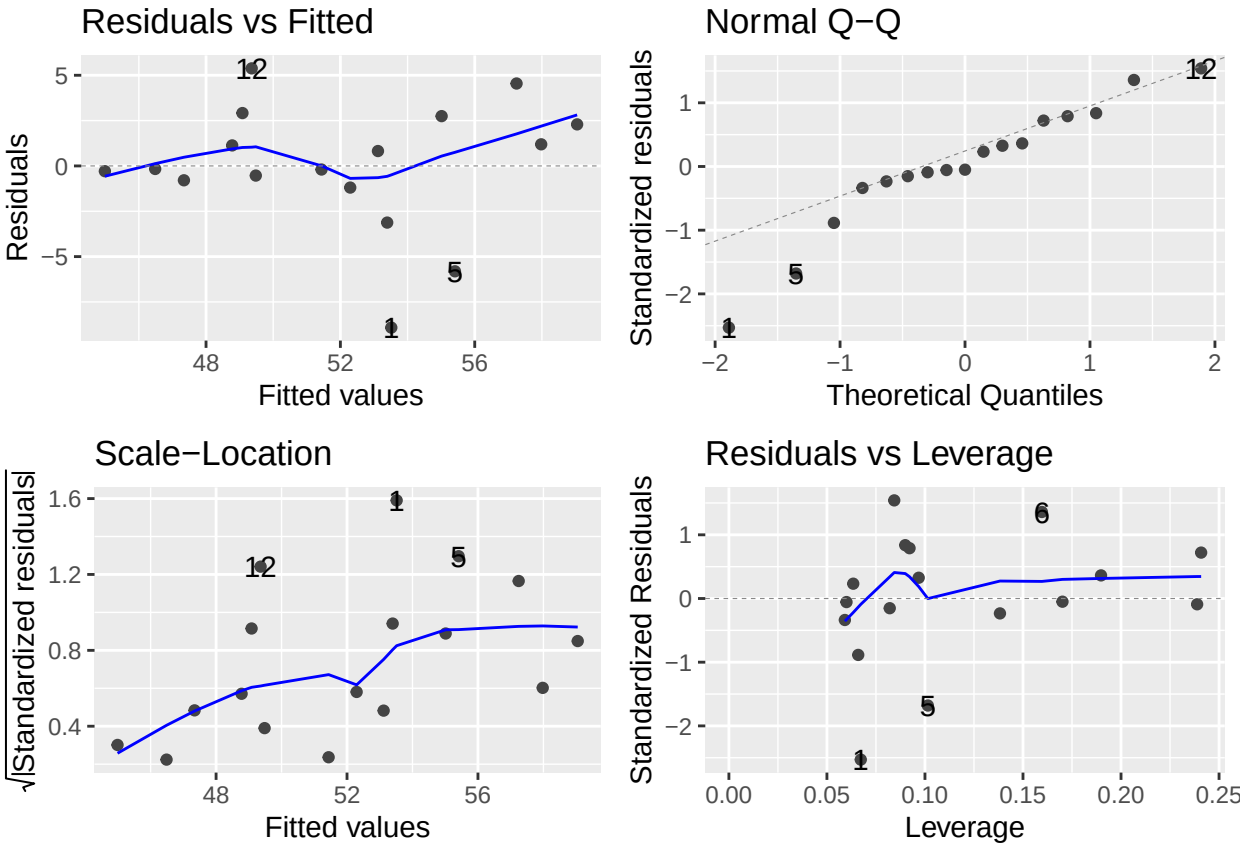

```
## And here's a "quantile-quantile plot"
qqnorm(fit$fitted.values)
```



For a prettier/faster version, you might have learned about the `autoplot()` function (part of the `ggfortify` library) from the R tutorial. It does some handy things, including "standardizing" the residuals to facilitate comparisons across disparate variable scales.

```
autoplot(fit)
```

```
## Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



PC6. Confidence interval for a coefficient

You might have done this by-hand using the formulas from the textbook or using R's built-in `confint()` function that I documented in the R tutorial. Here are examples of each:

First, by hand. You might recall the formula (provided on p. 334 of the textbook) for the confidence interval of a linear regression coefficient looks like this:

$$b_i \pm t^*_{df} \times SE_{b_i}$$

I can look up $t^*_{df=15}$ in the textbook table or calculate it directly before substituting into the equation.

```
t.star <- qt((1 - .05 / 2), 15)

beta <- coef(summary(fit))["growth", "Estimate"]
se <- coef(summary(fit))["growth", "Std. Error"]

## lower
beta - (t.star * se)
```

```
## [1] 1.617368
```

```
## upper
beta + (t.star * se)
```

```
## [1] 4.495312
```

Here's a much faster way to do that using the `confint()` command directly:

```
confint(fit, "growth", level = 0.95)
```

```
##           2.5 %   97.5 %
## growth 1.617368 4.495312
```

PC7. Out-of-sample prediction (interval)

Okay, so what share of the popular would we have expected Donald Trump to receive based solely on the performance of the economy over the past four years? Again, we can do this by hand using the formulas from the *OpenIntro* supplement or we can use some built-in R functions. Again, I'll demonstrate each.

First, by-hand. The point estimate for a predicted value is the fitted response obtained by entering the new predictor values into the regression equation from the model. In this case, that would look like the following:

$$\hat{y} = 46.18 + (3.05 * 2.5)$$

R can find that for me:

```
y.hat <- 46.18 + (3.05 * 2.5)
y.hat
```

```
## [1] 53.805
```

Now I can build the interval using this equation (from the supplement):

$$\hat{y} \pm t^*_{df} \times SE_{estimate}$$

I've already calculated $\hat{y}$ and t^*15. The formula for the standard error of the estimate (provided in the *OpenIntro* supplement) is:

$$SE_{estimate} = \sqrt{s_e^2 + \frac{s_e^2}{n} + (SE_{b_i})^2 \times (x^* - \bar{x})^2}$$

The *OpenIntro* supplement provides further explanation and definitions for each of the terms in this equation. I'll calculate and substitute the corresponding values here:

```
s.e.2 <- var(fit$residuals)
n <- length(hibbs$vote)
sq.dev <- (2.5 - mean(hibbs$growth))^2

## Standard error of the estimate:
se.est <- sqrt(s.e.2 + (s.e.2 / n) + ((se^2) * sq.dev))

## and the interval itself:
## lower
y.hat - (t.star) * se.est
```

```
## [1] 46.00746
```

```
## upper
y.hat + (t.star) * se.est
```

```
## [1] 61.60254
```

Phew. That was a lot. Let's do it easier with some built-in functions. Please note that in the R tutorial I neglected to document the `interval="prediction"` argument that can be passed to the `predict()` function. This makes it quite a bit easier to get what we want in this case:

```r
new.data <- data.frame(growth = 2.5)

## Here's the fitted value with the corresponding standard error:
pred <- predict(fit, new.data, interval = "prediction")

pred
```

```
##        fit      lwr      upr
## 1 53.82184 45.77162 61.87206
```

At this point, you might note that there are some very slight differences between the predicted values we calculated by hand versus those returned by the `predict()` function. Any guesses/suspicions where those differences might come from?

Statistical questions

SQ1. Interpret the results

In the analysis presented here I fit a regression model estimating incumbent party candidate share of the popular vote against per-capita economic growth for U.S. presidential elections since (approximately) the end of World War II (1952-2016).

The dependent variable, incumbent party candidate popular vote share, reflects raw proportions (rather than, say, share of electoral college votes, which is more directly linked to the electoral outcome) and takes continuous values between 44.6% and 61.8% ($\mu = 52\%$, $\sigma = 5.4$). The independent variable, per capital economic growth, is calculated as a proportion over the four years prior to the election in question. It is also continuous and ranges between -0.4% and 4.2% ($\mu = 1.9\%$, $\sigma = 1.4$). Both variables are distributed approximately normally and, as can be seen in the bivariate scatterplot, appear to have an appoximately linear relationship to each other. Given these characteristics (normality and bivariate linearity), I fit an ordinary least-squares regression model to estimate the association between economic growth and popular vote share. The model estimates a parameter value on economic growth against a null hypothesis of no association.

The model results indicate a good model fit overall ($F = 20.5$) that explains a substantial proportion of the variance in the outcome ($R^2 = 0.58$). The coefficient for growth ($\hat{b} = 3.06$, $\sigma = 0.68$) indicates that, on average, a 1% increase in per capita economic growth during the prior term is associated with a little more than a 3% increase in the incumbent party's share of the popular vote. The 95% confidence interval around this parameter estimate falls above zero and ranges between 1.6% and 4.5%, suggesting (together with the very small p-value returned by the formal null hypothesis test reported in the model summary) that the data provide strong support for a positive association between these two variables. Put in plainer terms, per capita economic growth during the preceding presidential term is associated with the incumbent party's popular vote share in U.S. presidential elections since World War II, and explains nearly 60% of the variation in this outcome.[1]

SQ2. Discuss regression diagnostics

As indicated above, the model diagnostics suggest that ordinary least squares (OLS, a.k.a., a linear model) is a reasonably good way to model the data here. The residuals appear normally distributed around zero (there is a slight left-skew induced by a few negative outliers) with approximately constant variance. Among the handful of outlying points, none seem to exert disproportionate leverage on the overall fit as illustrated in the QQ-plots and the Residuals vs. Leverage plot (this seems partly due to happenstance as the outliers sort of balance each other out). At the same time, further work might investigate the largest outlying points to try and understand why some incumbents might perform substantially worse-than-expected compared against

[1] We can talk about this in class, but this R^2 value is just bananas. By comparison, I've never encountered something so predictive of any outcome in any of the research settings I've worked in.

the baseline expectations informed by economic growth (more on that in Hibb's work and below when we talk about the out-of-sample prediction for 2020!). The fact that more of the largest residuals seem to fall below the fitted values indicates that there might be something else going on in these observations.

Another issue you might mention related to regression diagnostics concerns the (lack of) independence of the observations. The observations are not independent. They are all observations of an event occurring at regular intervals in the same geographic region of the world and many of them literally involve the same politicians across multiple elections. There actually *are* some analytic approaches to addressing some of these issues via adjustments to the standard errors and/or more complex modeling approaches. We can discuss some of these in class.

SQ3. Correlation vs. covariance vs. OLS

Okay, so there are a bunch of things you could say here. I'll stick to some basic points:

1. All three statistical procedures support the idea of a positive relationship (association) between these two variables.
2. Some differences between them include: Correlation scales the association to a value between -1 and 1; Covariance reflects the association in terms of the respective spreads of the two distributions; the OLS parameter captures the association in terms of the underlying measures (percentages) and the R^2 value reflects the proportion of the variation in the outcome explained by the predictor.
3. Depending on exactly what you're trying to say about the relationship, each of the three can provide distinct information. Personally, I prefer the regression model results because I can use them to generate confidence and predictive intervals in terms that are quite easy to interpret directly.

SQ4. Interpret out-of-sample prediction

Trump seems to have under-performed the model-based prediction by a large, but not unpredictable or unprecedented degree. While the margin of his popular vote loss seems likely to grow as absentee and other mail-in ballots continue to be counted and certified, the currently observed popular vote share of 47.6% for an incumbent party candidate falls within the 95% prediction interval we estimate here. Also, it is worth noting that Trump's vote proportion is very much in-line with his previous performance in 2016 (46.1% of the popular vote in that cycle). We can further consider Trump's popular vote share in historical perspective by comparing his relative (under)performance against previous under-performers: Adlai Stevenson in 1952 and Hubert Humphrey in 1968. The model-based residual (error) for the fitted values in these cases is -8.9% and -5.8% respectively (compared against Trump's approximate residual of -6.2%). As Hibbs notes in his initial work on this data, both the Stevenson and Humphrey under-performances came against backdrops of bloody wars with large-scale losses of American soldiers' lives (in Korea and Vietnam respectively).

The U.S. has not experienced comparable large-scale fatalities in any current military conflicts, so what could explain the relative under-performance in Trump's case? While some might point to Trump's apparent disdain for democratic political institutions or frequent violations of widespread American cultural norms (e.g., his encouragement of White Nationalist terrorist/militia groups), previous incumbent party candidates (and/or their followers) have embraced similar domestic political and cultural positions (e.g., Nixon, Reagan) and have *not* underperformed to the same degree as Trump. Indeed, the factor that differentiates Trump from these predecessors and appears to far more closely resemble the impact of a deadly war has been the COVID-19 pandemic.

SQ5. Revisit theory

The theory, to the extent that I described it in the problem set preface, is that economic growth during the preceding presidential term helps to explain incumbent party popular vote share in U.S. presidential elections since World War II. The popular vote outcome of President Trump in 2020 falls below the point estimate, but within the 95% prediction interval predicted from a model fitted on 1952-2016 electoral outcomes. Trump's relative under-performance is, in this sense, unsurprising in the context of the model. His share of the popular vote is aligned with his performance in the most recent presidential election and is not the largest deviation

from the model fitted values in the dataset. Another possible explanation for his relatively low vote share as an incumbent is the COVID-19 pandemic, an ongoing mass-casualty event that seems likely to wind up causing more deaths of U.S. citizens than all the wars of the 20th century combined and that began early in the final year of his term. Given that mass-casualty events are the other variable in Hibbs' full model, this out-of-sample observation seems to fit with Hibbs' theory quite well, although you might expand his idea of "peace" to include something like "public health" or "absence of large-scale death."