Lab 3

Alexander Baxter - edited, Samuel D. Aragones

Winter 2024

Objectives

1. One Way ANOVA with Dummy Coding
2. Interactions with Dummy Coding
3. ANOVA with Effect Coding (Supplementary Information)

Load required packages

```r
library(car)       # Has data we will use
library(ggplot2)   # Graphing
library(scales)    # Graph formatting
library(dplyr)     # Data cleaning and handling
library(ggpubr)    # Multi-Panel Graphs
library(broom)     # Data cleaning
library(knitr)     # Tables
library(flextable) # Tables
library(stargazer) # Tables
```

For today's lab, we will be using the following data sets:

- `Leinhardt` (from the `car` package, contains data on income, infant mortality, and oil production)
- `Salaries` (a built in data set, contains data on professor salaries)

```r
Leinhardt3 <- Leinhardt[Leinhardt$region != 'Europe', ]
# We will use this data frame without Europe in it in an exammple below
```

If you care to know more about the data sets:

```r
?Leinhardt
?Salaries
```

Dummy Coding

Dichotomous Variables (Review)

Dummy coding is one of several methods you can use to enter categorical variables into regression analyses. Dummy coded variables take on values of 0 and 1. The group coded as 0 is the reference group, and all estimates for the group(s) coded as 1 will be for differences relative to the reference group.

Thus far in the class, we have used dummy codes to enter dichotomous variables into analyses (for example, sex or diagnosis status). For example, if we wanted to test differences between male and female professor’s salaries, we could do a linear regression, with sex as a dummy coded variable.

```
# First, create a dummy coded variable.
# In this case, "Female" is the reference group (0)

Salaries$sex.code <- ifelse(Salaries$sex == "Female", 0, 1)

# Second, fit the linear model

m1 <- lm(salary ~ sex.code, data = Salaries)
summary(m1)
```

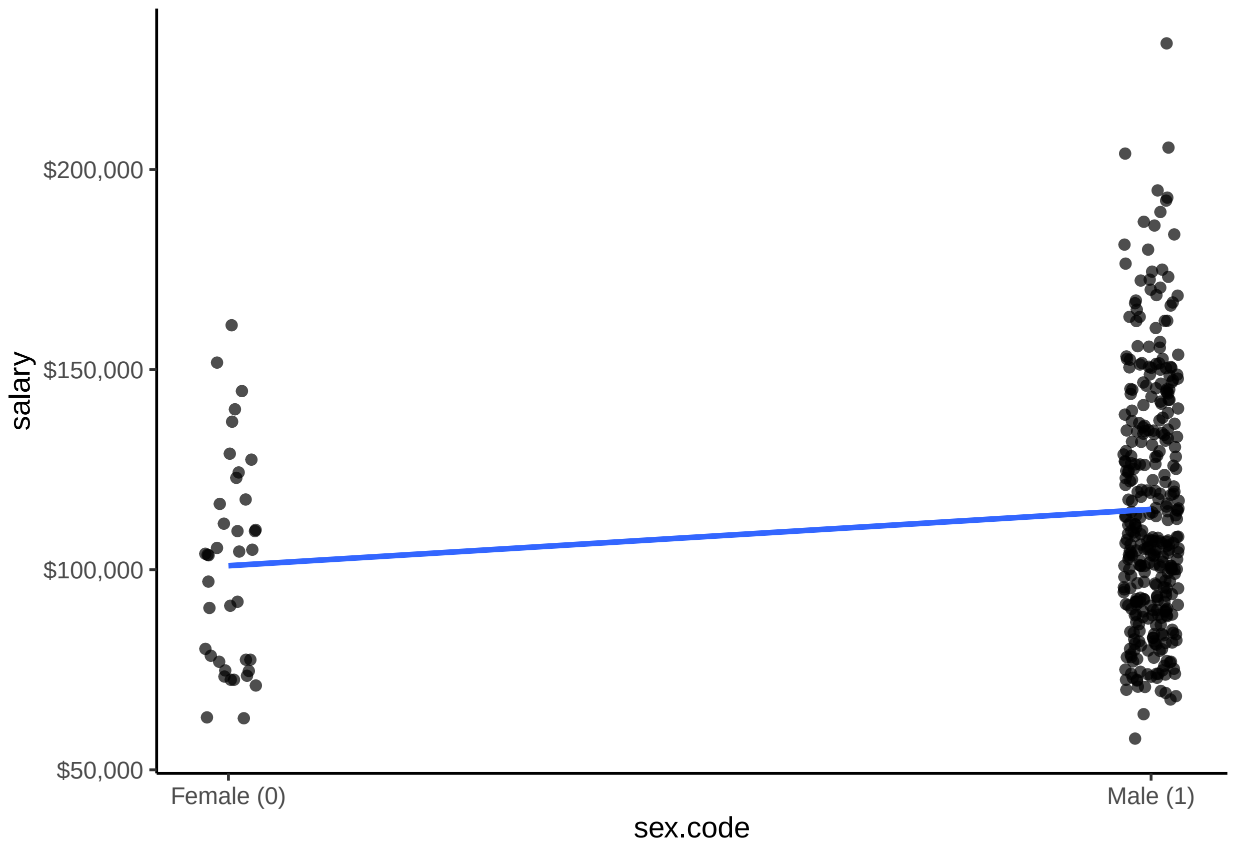
```
##
## Call:
## lm(formula = salary ~ sex.code, data = Salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57290 -23502  -6828  19710 116455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   101002       4809  21.001  < 2e-16 ***
## sex.code       14088       5065   2.782  0.00567 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 30030 on 395 degrees of freedom
## Multiple R-squared:  0.01921,    Adjusted R-squared:  0.01673
## F-statistic: 7.738 on 1 and 395 DF,  p-value: 0.005667
```

$$\widehat{\texttt{salary}}_i = 101002 + 14088\texttt{sex}_i$$

- The **Intercept**, 101002, is the predicted salary for females (i.e., when *sex* is 0).
- The **Estimate for sex.code**, 14088, is the predicted increase in *salary* for a 1-unit increase in sex (i.e., going from females’ mean to males’ mean). Because Males were coded as 1, this means that males’ predicted salary is: 101002 + 14088 = 115090.

```
ggplot(data = Salaries,
       aes(x = sex.code, y = salary)) +
  geom_jitter(width = 0.03,
              alpha = 0.7) +
  geom_smooth(method = "lm", se = F) +
  scale_x_continuous(breaks = c(0,1),
                     labels = c("Female (0)", "Male (1)")) +
  scale_y_continuous(label = dollar) +
  theme_classic()
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



Notice what happens when we change the reference group to male.

```
# In this case, "Male" is the reference group (0)
Salaries$sex.code_v2 <- ifelse(Salaries$sex == "Male", 0 ,1)

# Second, fit the linear model
m2 <- lm(salary ~ sex.code_v2, data = Salaries)
summary(m2)
```

```
##
## Call:
## lm(formula = salary ~ sex.code_v2, data = Salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57290 -23502  -6828  19710 116455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   115090       1587  72.503  < 2e-16 ***
## sex.code_v2   -14088       5065  -2.782  0.00567 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 30030 on 395 degrees of freedom
## Multiple R-squared:  0.01921,    Adjusted R-squared:  0.01673
## F-statistic: 7.738 on 1 and 395 DF,  p-value: 0.005667
```

$$\widehat{\texttt{salary}}_i = 115090 - 14088\texttt{sex}_i$$

Lets compare the two regression analyses in a table.

There are a few different ways to display table outputs in R Markdown files. The easiest way is with `kable`. However, if you are using a dark-theme and working in Visual mode, these tables don't always display very well when you are working in the Markdown editor. So another option to use is the `flextable` package. Below I will show how both options look. For the remainder of the lab, I will use flextable (if you want to read more about these functions, check out this link). However, you are not expected to use these functions if you do not want to, you can use `kable` for your assignments.

```r
# Puts model information into a nice format
tidy_m1 <- tidy(m1)
tidy_m2 <- tidy(m2)


# bind_rows() combines our tidy output from above
# .id specifies a column to differentiate between M1 and M2
df <-
  bind_rows(
    M1 = tidy_m1,
    M2 = tidy_m2,
    .id = "Model")
```

With kable or stargazer:

```r
kable(df,
      digits = 2,
      col.names = c("Model", "Term", "Estimate", "S.E.", "Statistic", "p")) #kable reads more as if it
```

Model	Term	Estimate	S.E.	Statistic	p
M1	(Intercept)	101002.41	4809.39	21.00	0.00
M1	sex.code	14088.01	5064.58	2.78	0.01
M2	(Intercept)	115090.42	1587.38	72.50	0.00
M2	sex.code_v2	-14088.01	5064.58	-2.78	0.01

```r
stargazer(m2, type='text', header=FALSE)
```

```
## 
## ===============================================
##                         Dependent variable:    
##                     ---------------------------
##                               salary           
## -----------------------------------------------
## sex.code_v2              -14,088.010***        
##                            (5,064.579)         
##                                                
## Constant                 115,090.400***        
##                              (1,587.378)
##
## -----------------------------------------------
## Observations                    397
## R2                             0.019
## Adjusted R2                    0.017
## Residual Std. Error    30,034.610 (df = 395)
## F Statistic           7.738*** (df = 1; 395)
## ===============================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

With flextable

```r
df %>% #uses more tidyverse type language for construction of tables
  flextable() %>%
  colformat_double(j = 3:5, digits = 1) %>%
  colformat_double(j = 6, digits = 3) %>%
  merge_v(j = 1) %>%
  fix_border_issues() %>%
  theme_vanilla()
```

Model	**term**	**estimate**	**std.error**	**statistic**	**p.value**
M1	(Intercept)	101,002.4	4,809.4	21.0	0.000
	sex.code	14,088.0	5,064.6	2.8	0.006
M2	(Intercept)	115,090.4	1,587.4	72.5	0.000
	sex.code_v2	-14,088.0	5,064.6	-2.8	0.006

Returning to the example: Lets also look at the R Squared values.

```r
# R Squared for Model 1
summary(m1)$r.squared
```

```
## [1] 0.01921278
```

```r
# R Squared for Model 2
summary(m2)$r.squared
```

```
## [1] 0.01921278
```

Important take-away points:

- The intercept changed. This is because we changed what 0 means (Female or Male). This number represents the estimated mean of y for the group that is coded as 0 (the reference group).
- The absolute estimate of *sex.code* did not change, but it did change signs. This is because the the estimate represents the predicted difference between the two groups, relative to the group that is 0. So, when Females were the reference group, the estimate of *sex.code* was positive because Males had a higher mean. When Males were the reference group, the estimate was negative, because Females had a lower mean than males.

- The R^2 did not change. This is because changing the reference group does not change the linear relationship in the data, or the difference between the groups. It just changes where the model will start.
- Notice that in each case, the intercepts and slopes correspond to the observed means

```
Salaries %>%
  group_by(sex) %>%
  summarise(mean_salary = mean(salary)) %>%
  flextable() %>%
  colformat_double(digits = 1)
```

sex	mean_salary
Female	101,002.4
Male	115,090.4

- Model 1 (with "Female" as the reference group)
  - Intercept = 101002
  - Intercept + Slope = 101002 + 14088 = 115090
- Model 2 (with "Male" as the reference group)
  - Intercept $= 115090 $
  - Intercept + Slope = 115090 − 14088 = 101002

Automatic Dummy Coding with Factors

In the previous example, we saw how to manually code the dummy coded variable, and use that in the regression model (i.e., with `sex.code`).

However, as long as the categorical variable (in this case, `sex`) is coded as a factor, you can use it in the linear model. If you do this, the `lm()` function will automatically dummy code your variable, and will use the first level as the reference group.

Lets check how *sex* is coded right now.

```
class(Salaries$sex) # It should be a factor already
```

```
## [1] "factor"
```

Lets try it and see what happens when we use `sex` as the predictor in the model instead of `sex.code`. Remember, `sex` is coded as "Male" and "Female".

```
summary(lm(salary ~ sex, data = Salaries))
```

```
##
## Call:
## lm(formula = salary ~ sex, data = Salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57290 -23502  -6828  19710 116455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   101002       4809  21.001  < 2e-16 ***
## sexMale        14088       5065   2.782  0.00567 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30030 on 395 degrees of freedom
## Multiple R-squared:  0.01921,    Adjusted R-squared:  0.01673
## F-statistic: 7.738 on 1 and 395 DF,  p-value: 0.005667
```

Notice that the estimate for sex is listed as `sexMale`. This means that the slope represents "the effect of being Male for sex". In other words, Female was coded as the reference group. In this case, the analysis automatically coded Males as 1 and Females as 0.

Why were females coded as the reference group and not males?

It has to do with the orders of the levels for the factor. The first level will automatically be coded as 0, the second level will automatically be coded as 1.

To check the levels of a factor, you can use `levels()`.

```r
levels(Salaries$sex) # Female is the first level
```

```
## [1] "Female" "Male"
```

This function returns the levels of the factor, in order. By default, R orders factors alphanumerically; hence Female is the first level and Male is the second.

To change the order of levels, use the `factor()` function and set the levels with the `levels` argument. Lets create a new variable in the data called `sex2` that has Male coded as the first level and Female coded as the second level.

```r
Salaries$sex2 <-
  factor(Salaries$sex,               # The original variable
         levels = c("Male","Female") # The desired order of the levels
         )

levels(Salaries$sex2)
```

```
## [1] "Male"   "Female"
```

Note: Whatever you enter for the levels argument must match the data values exactly as they exist in the original variable, otherwise R will turn any value that is not listed in levels into an NA value. For example, if I did not use a capital F for "Female", all values of "Female" would have been turned to NA.

```r
# We could also use the relevel() function to set our reference group, although
# this doesn't let you set the order of the remaining levels in the case of 3+
# levels. So I don't really recommend this option. Just an FYI that this exists.

Salaries$sex2 <- relevel(Salaries$sex, ref = "Male")
```

Now, lets try using `sex2` in the regression model.

```
summary(lm(formula = salary ~ sex2, data = Salaries))
```

```
##
## Call:
## lm(formula = salary ~ sex2, data = Salaries)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57290 -23502  -6828  19710 116455
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   115090       1587  72.503  < 2e-16 ***
## sex2Female    -14088       5065  -2.782  0.00567 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30030 on 395 degrees of freedom
## Multiple R-squared:  0.01921,    Adjusted R-squared:  0.01673
## F-statistic: 7.738 on 1 and 395 DF,  p-value: 0.005667
```

Now, Male is the reference group, and the slope estimate is for the Female group.

Comparison with t-test

Lets compare the *t*-values from the regression equations with male and female reference groups. Remember, the *t*-values from a simple regression are calculated by dividing the slope estimate by the SE of the estimate, and then are used to determine the significance of the slope (relative to the t distribution with the associated DF from the analysis).

Now, lets run an independent (Student's) *t*-test, and see how it compares to the *t*-statistics we got from our regression.

```
t_test <- t.test(salary ~ sex, data = Salaries, var.equal = TRUE)
t_test
```

```
##
##  Two Sample t-test
##
## data:  salary by sex
## t = -2.7817, df = 395, p-value = 0.005667
## alternative hypothesis: true difference in means between group Female and group Male is not equal to
## 95 percent confidence interval:
##  -24044.910  -4131.107
## sample estimates:
## mean in group Female   mean in group Male
##             101002.4             115090.4
```

```
# t-test
t_test$statistic
```

```
##         t
## -2.781674
```

```r
# Regression with female reference group
tidy_m1$statistic[2]
```

```
## [1] 2.781674
```

```r
# Regression with male reference group
tidy_m2$statistic[2]
```

```
## [1] -2.781674
```

Dummy Coding with 3 Levels (One-Way ANOVA)

Thus far we have only discussed dummy codes with only two levels. But what if we had three groups?

When dummy coding a variable, you will have $J - 1$ columns for the dummy codes, where J is the number of groups in the categorical data. Hence, for only 2 groups, there is 2 - 1 = 1 column.

Lets say we have a variable with 3 groups. This means we will have 2 columns of dummy codes. Each column with be coded with 0s and 1s. The rows belonging to the reference group will have 0s in both columns. Each column will then have 1s coded for the second and third group.

Lets demonstrate this with the `Leinhardt3` dataset. We will predict infant mortality rate (*infant*) based on the region of each country (Americas, Asia, or Africa).

First, you need to choose a reference group. Let's say the Americas will be our reference group.

Second, we need to create 2 new columns: one for "The Region is in Asia" ($Asia$) and one for "The Region is in Africa" ($Africa$).

For the $Asia$ variable:

- $Asia = 1$ if region is Asia
- $Asia = 0$ if region is not Asia (i.e., Americas or Africa)

For the $Africa$ variable:

- $Africa = 1$ if region is Africa
- $Africa = 0$ if region is not Africa (i.e., Americas or Asia)

```r
# To create the dummy codes
Leinhardt3$Africa <- ifelse(Leinhardt3$region =="Africa", 1, 0)
Leinhardt3$Asia <- ifelse(Leinhardt3$region == "Asia", 1, 0)
```

Here is a summary of the coding we just did:

```r
Leinhardt3 %>%
  group_by(region) %>%
  summarise(africa = unique(Africa),
            asia = unique(Asia)) %>%
  flextable() %>%
  colformat_double(digits = 1)  %>%
  merge_v(j = 1) %>%
  fix_border_issues() %>%
  theme_vanilla()
```

region	africa	asia
Africa	1.0	0.0
Americas	0.0	0.0
Asia	0.0	1.0

Now, we can add our newly created dummy variables to predict infant morality rates for different regions:

```
m3 <-  lm(infant ~ Africa + Asia, data = Leinhardt3)
summary(m3)
```

```
##
## Call:
## lm(formula = infant ~ Africa + Asia, data = Leinhardt3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -43.23  -9.12  18.96 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    55.12      18.65   2.956 0.004090 **
## Africa         87.17      23.93   3.643 0.000477 ***
## Asia           41.05      25.12   1.634 0.106154
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 87.46 on 80 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.124
## F-statistic: 6.803 on 2 and 80 DF,  p-value: 0.001869
```

$$\widehat{\texttt{infant}}_i = 55.12 + 87.17\texttt{africa}_i + 41.05\texttt{asia}_i$$

Interpretations:

- The *Americas* are the reference here (i.e., when *Africa* and *Asia* are both equal to 0, this is the expected infant mortality for the *Americas*). So the average infant mortality rate for regions in the *Americas* is 55.12 per 1,000 live births.
  - This corresponds to $55.12 + (87.17 * 0) + (41.05 * 0)$
  - Conceptually, this equation is the same interpretation that we have been using for the intercept all along: The intercept is the predicted value of y when all predictors are 0. I.e., the predicted infant mortality rate is 55.12 when the dummy codes for *Asia* and *Africa* are 0. What does it mean for these dummy codes to be 0? If both values are 0, that corresponds to *Americas*.

- The slope for *Africa* refers to the difference between the mean value for Africa and the mean value for the Americas (the reference group). This difference is significant at the 0.05 level ($p = .0005$).
  - This corresponds to $55.12 + (87.17 * 1) + (41.05 * 0) = 142.29$
  - Hence, the average infant mortality rate for African regions is 142.29 (per 1,000 live births).
- The slope for *Asia* refers to the difference between the mean value for Asia and the mean value for the Americas (the reference group). This mean difference is not significantly different from 0 ($p = .11$).
  - This corresponds to $55.12 + (87.17 * 0) + (41.05 * 1) = 96.17$
  - The average infant mortality rate for Asian regions is 96.17 (per 1,000 live births).

Now lets compare these interpretations to the actual means of the regions:

```
Leinhardt3 %>%
  group_by(region) %>%
  summarise(mean_infant_mortality = mean(infant, na.rm = T)) %>%
  flextable() %>%
  colformat_double(digits = 1)
```

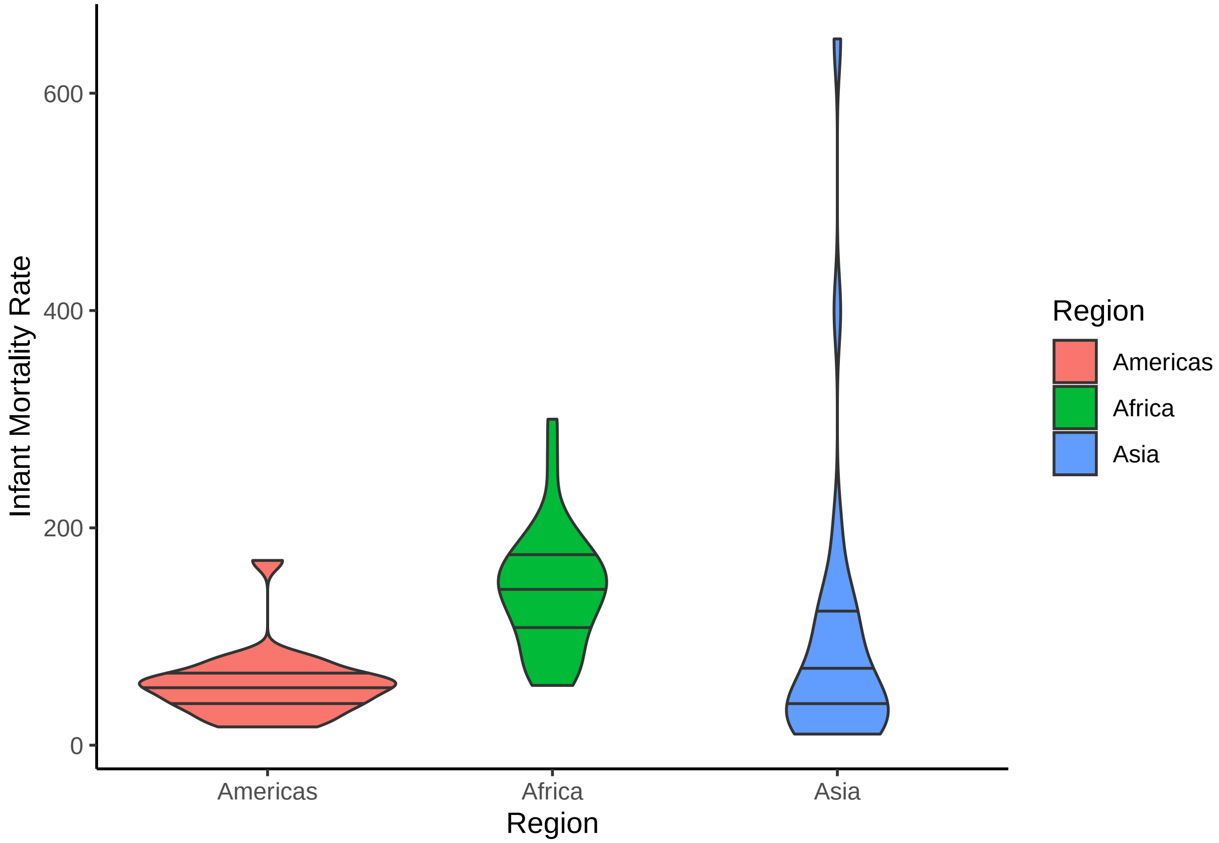
region	mean_infant_mortality
Africa	142.3
Americas	55.1
Asia	96.2

One way we could visualize the mortality rates between regions is with a violin plot like this:

```
# Setting levels in the order I want them graphed
Leinhardt3$region <-
  factor(Leinhardt3$region,
         levels = c("Americas", "Africa", "Asia"))
  ## It is helpful to put the reference group first

# The graph
ggplot(data = Leinhardt3,
       aes(x = region, y = infant)) +
  geom_violin(aes(fill = region),
              draw_quantiles = c(0.25, 0.5, 0.75)) +
  labs(x = "Region",
       y = "Infant Mortality Rate",
       fill = "Region") +
  theme_classic()
```

```
## Warning: Removed 4 rows containing non-finite values (‘stat_ydensity()‘).
```



We could also graph the means on top of the "violins"

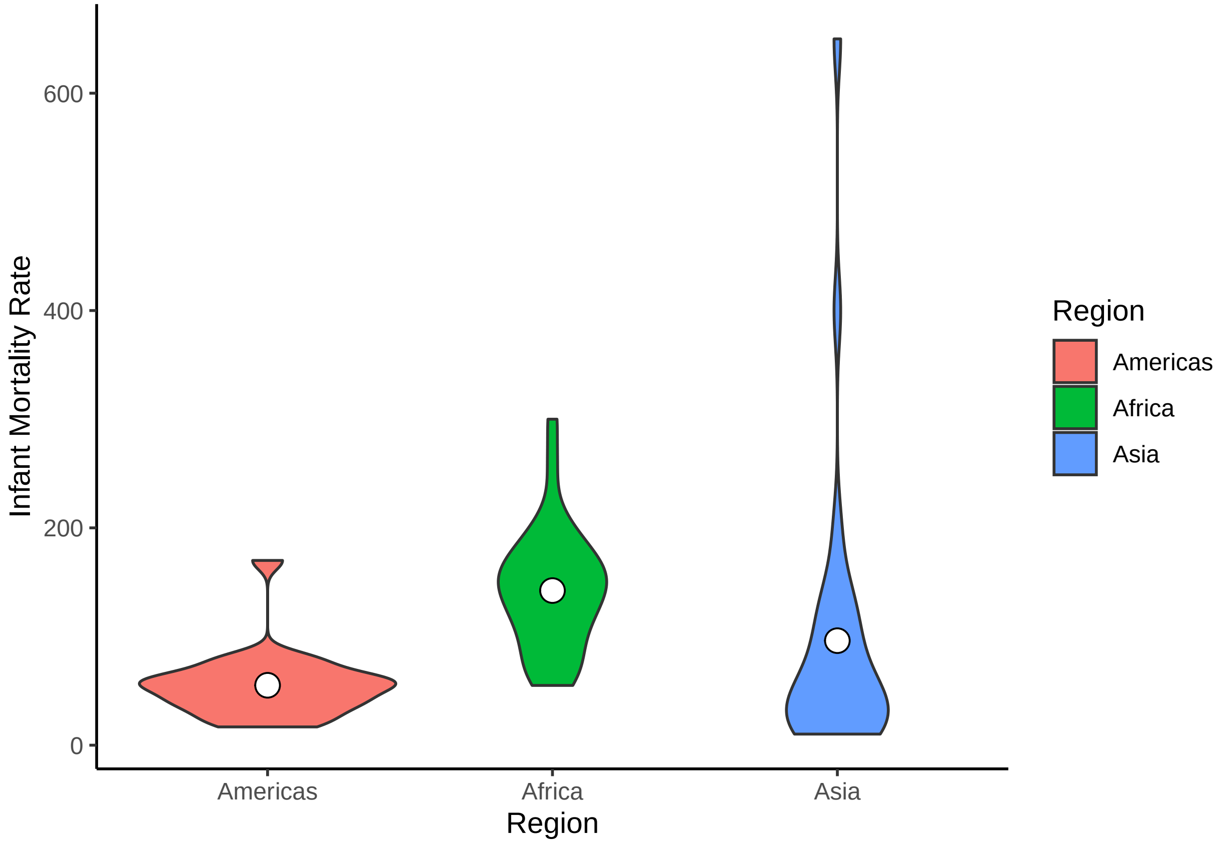
```
ggplot(data = Leinhardt3,
       aes(x = region,y = infant)) +
  geom_violin(aes(fill = region)) +

  # stat summary takes our "y" variable and applies a "fun" (function) to it
  stat_summary(fun = mean,
               geom = "point",
               shape = 21,
               col = "black",
               fill = "white",
               size = 4) +

  labs(x = "Region",
       y = "Infant Mortality Rate",
       fill = "Region") +
  theme_classic()
```

```
## Warning: Removed 4 rows containing non-finite values (‘stat_ydensity()‘).
```

```
## Warning: Removed 4 rows containing non-finite values (‘stat_summary()‘).
```



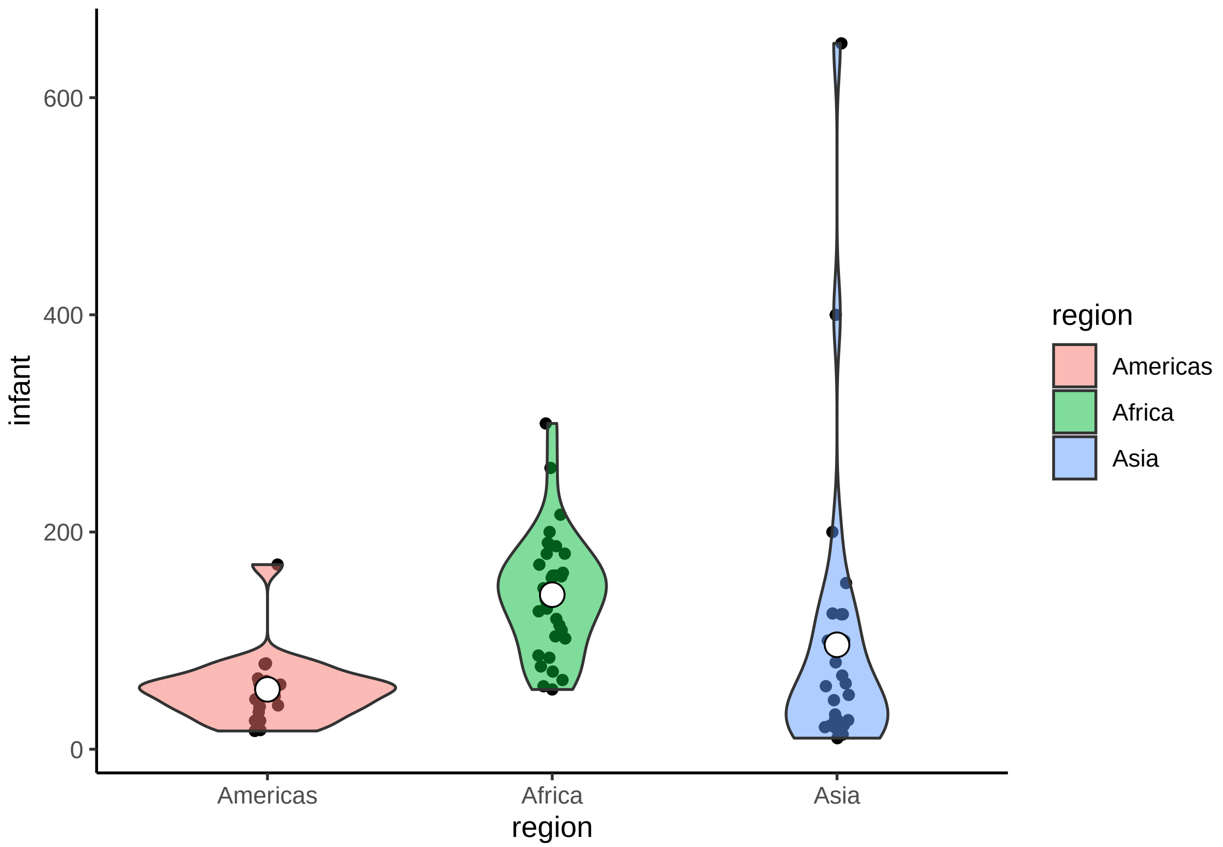
Something that could be nice is to further add the actual data points underneath the violins

```
ggplot(data = Leinhardt3,
       aes(x = region,y = infant)) +
  geom_jitter(width = 0.05,
              alpha = 1) +
  geom_violin(aes(fill = region),
              alpha = 0.5) +
  stat_summary(fun = mean,
               geom = "point",
               shape = 21,
               col = "black",
               fill = "white",
               size = 4) +
  theme_classic()
```

```
## Warning: Removed 4 rows containing non-finite values (‘stat_ydensity()‘).

## Warning: Removed 4 rows containing non-finite values (‘stat_summary()‘).

## Warning: Removed 4 rows containing missing values (‘geom_point()‘).
```

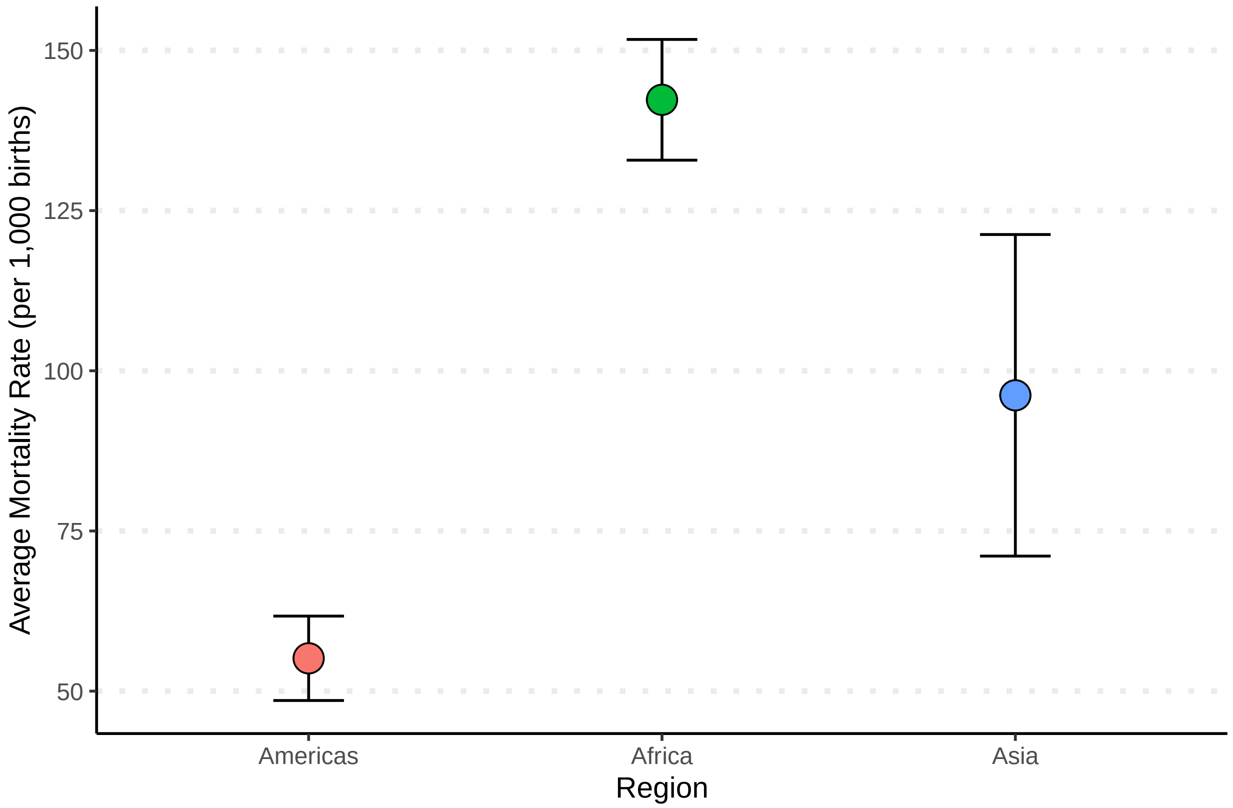


A common way to visualize this type of analysis is to plot the means with +/- 1 SE bars, like this:

```
leinhardt_means <-
  Leinhardt3 %>%
  group_by(region) %>%
  summarise(mean = mean(infant, na.rm = T),
            n = length(infant),
            sd = sd(infant, na.rm = T),
            se = sd / sqrt(n),
            ub = mean + se, # Upper Boundary of SE Bar
            lb = mean - se  # Lower Boundary of SE Bar
            )

ggplot(data = leinhardt_means,
       aes(x = region, y = mean, fill = region)) +
  geom_errorbar(aes(ymin = lb, ymax = ub),
                width = 0.2) +
  geom_point(size = 5, shape = 21) +
  labs(x = "Region",
       y = "Average Mortality Rate (per 1,000 births)",
       caption = "Note. Error bars denote +/- 1 SE") +
  theme_classic() +
  theme(panel.grid.major.y = element_line(linetype = "dotted", size = 1),
        legend.position = "none")
```

```
## Warning: The ‘size‘ argument of ‘element_line()‘ is deprecated as of ggplot2 3.4.0.
## i Please use the ‘linewidth‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Note. Error bars denote +/− 1 SE

Comments on Significance:

Lets return again to the model, and interpret what is and is not significant (at $p < .05$). For reference, here is the model again:

```
summary(m3)
```

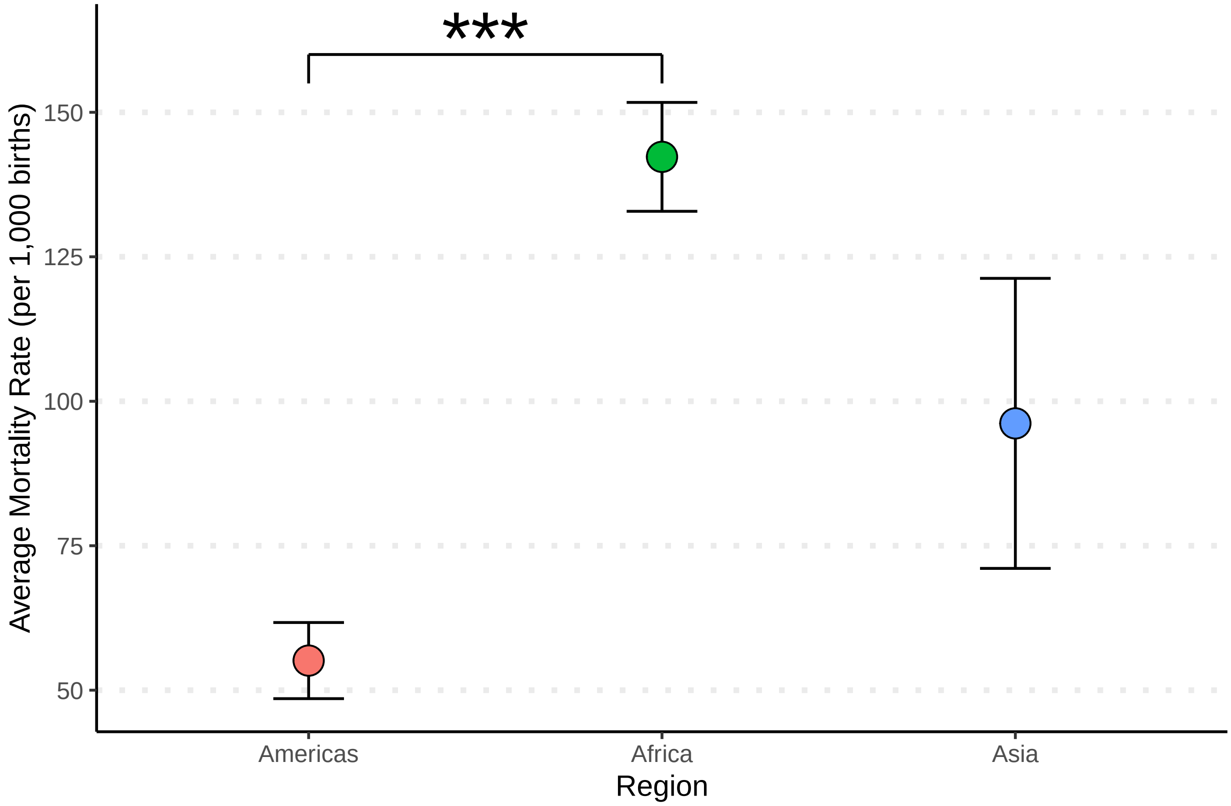
```
##
## Call:
## lm(formula = infant ~ Africa + Asia, data = Leinhardt3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -43.23  -9.12  18.96 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    55.12      18.65   2.956 0.004090 **
## Africa         87.17      23.93   3.643 0.000477 ***
## Asia           41.05      25.12   1.634 0.106154
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 87.46 on 80 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.124
## F-statistic: 6.803 on 2 and 80 DF,  p-value: 0.001869
```

- The overall model is significant ($F(2, 80) = 6.80$, p = .002). This indicates that *region* has an effect on *infant* mortality.
  - Remember, the F test is an omnibus test (an "overall test"), which means we don't know where the effect is or what direction the effect is. We just know that there is an effect.
- The intercept (the mean of the reference group, *Americas*) is significantly different from 0.
  - Note: this may or may not be pertinent or interesting depending on the hypothesis you are testing; in our case this is not that interesting, because it is not possible to have negative infant mortality, so by definition all regions' infant mortality should be greater than 0.
- The difference between the mean of *Africa* and the mean of *Americas* is significantly greater than 0 ($b = 87.17$; $p < .001$). In other words, the mean of *Africa* is significantly higher than the mean of *Americas*.
  - We know the direction of the effect because the b slope is positive (87.17).
- The difference between the mean of *Asia* and the mean of *Americas* is not significantly different from 0 ($b = 41.05$; $p = .11$). In other words, the means of *Asia* and the *Americas* are not significantly different.
- We cannot say anything about the difference between the means of *Africa* and *Asia*, as all comparisons are made relative to the reference group (*Americas*).

If you wanted to indicate significant comparisons, you could do so with a series of annotations. For example:

```r
ggplot(data = leinhardt_means,
       aes(x = region, y = mean, fill = region)) +
  geom_errorbar(aes(ymin = lb, ymax = ub), width = 0.2) +
  geom_point(size = 5, shape = 21) +
  theme_classic() +
  labs(x = "Region",
       y = "Average Mortality Rate (per 1,000 births)",
       caption = "Note. Error bars denote +/- 1 SE") +
  theme_classic() +
  theme(panel.grid.major.y = element_line(linetype = "dotted", size = 1),
        legend.position = "none") +

  # draws long horizontal line
  annotate(geom = "segment",
           x = 1, xend = 2, y = 160, yend = 160) +
  # draws vertical lines
  annotate(geom = "segment",
           x = 1, xend = 1, y = 160, yend = 155) +
  annotate(geom = "segment",
           x = 2, xend = 2, y = 160, yend = 155) +
  # add an asterisk
  annotate(geom = "text", label = "***",
           x = 1.5, y = 163, size = 10, color = "black")
```



Note. Error bars denote +/− 1 SE

You could also make a table:

```
tidy_m3 <- tidy(m3)
colnames(tidy_m3) <- c("Term", "Estimate", "S.E.", "t", "p") # Cleaning col names

tidy_m3  %>%
  flextable() %>%
  colformat_double(j = 1:4, digits = 1) %>%
  colformat_double(j = "p", digits = 3)
```

Term	Estimate	S.E.	t	p
(Intercept)	55.1	18.6	3.0	0.004
Africa	87.2	23.9	3.6	0.000
Asia	41.0	25.1	1.6	0.106

Let's compare the regression analysis we did to an ANOVA. Remember, in R we can do ANOVA by using the `aov` function.

```
# ANOVA:
summary(aov(infant ~ region, data = Leinhardt3))
```

```
##             Df Sum Sq Mean Sq F value  Pr(>F)
## region       2 104061   52031   6.803 0.00187 **
## Residuals   80 611878    7648
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 4 observations deleted due to missingness
```

```
# Regression with Dummy Codes:
summary(m3)
```

```
##
## Call:
## lm(formula = infant ~ Africa + Asia, data = Leinhardt3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -43.23  -9.12  18.96 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    55.12      18.65   2.956 0.004090 **
## Africa         87.17      23.93   3.643 0.000477 ***
## Asia           41.05      25.12   1.634 0.106154
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 87.46 on 80 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.124
## F-statistic: 6.803 on 2 and 80 DF,  p-value: 0.001869
```

Notice that the F-statistic is the same in both analyses. However, we get slightly different information from the ANOVA vs the regression.

Remember, the ANOVA is an omnibus test; it does not tell us which groups are significantly different. It just tells us that overall there is an effect. That is why there is only an F table given.

The regression with dummy coded variables gives us the same information, plus a little more. It lets us compare one group to the other two, one at a time (in this case, Americas vs Africa and Americas vs Asia). The regression we did does not tell us whether Africa & Asia are significantly different.

Dummy Coding: The reference group matters!

To demonstrate what happens if we change the reference group, lets re-do the regression analysis above, but this time Asia will be the reference group.

```
Leinhardt3$Africa_v2 <- ifelse(Leinhardt3$region == "Africa", 1, 0)
Leinhardt3$Americas_v2 <- ifelse(Leinhardt3$region == "Americas", 1, 0)

m3_Asia <- lm(infant ~ Africa_v2 + Americas_v2, data = Leinhardt3)
summary(m3_Asia)
```

```
##
## Call:
```

```
## lm(formula = infant ~ Africa_v2 + Americas_v2, data = Leinhardt3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -43.23  -9.12  18.96 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    96.17      16.83   5.714 1.82e-07 ***
## Africa_v2      46.12      22.54   2.046   0.0441 *
## Americas_v2   -41.05      25.12  -1.634   0.1062
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 87.46 on 80 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.124
## F-statistic: 6.803 on 2 and 80 DF,  p-value: 0.001869
```

Now, lets compare how the parameters change when the reference group is changed:

```r
bind_rows(m3_Americas = tidy(m3),
          m3_Asia = tidy(m3_Asia),
          .id = "Model") %>%
  flextable() %>%
  merge_v(j = 1) %>%
  fix_border_issues() %>%
  theme_vanilla() %>%
  colformat_double(digits = 2) %>%
  autofit()
```

Model	term	estimate	std.error	statistic	p.value
m3_Americas	(Intercept)	55.12	18.65	2.96	0.00
	Africa	87.17	23.93	3.64	0.00
	Asia	41.05	25.12	1.63	0.11
m3_Asia	(Intercept)	96.17	16.83	5.71	0.00
	Africa_v2	46.12	22.54	2.05	0.04
	Americas_v2	-41.05	25.12	-1.63	0.11

```r
bind_rows(m3_Americas = glance(m3),
          m3_Asia = glance(m3_Asia),
          .id = "Model")  %>%
  select(Model,
         R.2 = r.squared,
         p = p.value,
         df,
         logLik,
         AIC,
         BIC,
         dev = deviance,
         df.resid = df.residual,
         n = nobs
         ) %>%
  flextable() %>%
  colformat_double(digits = 2) %>%
  colformat_double(j = "p", digits = 3) %>%
  autofit()
```

Model	R.2	p	df	logLik	AIC	BIC	dev	df.resid
m3_Americas	0.15	0.002	2.00	-487.35	982.70	992.37	611,877.80	80
m3_Asia	0.15	0.002	2.00	-487.35	982.70	992.37	611,877.80	80

```
## Note: glance extracts a few key parameters from regression model objects. Handy!
## For the sake of space I only selected a handful of the available values.
```

- The intercept changed because we changed what 0 means by changing the reference group. Hence the intercept changes, as it represents the mean of the reference group.
- The slopes also change, because they represent the difference between the group mean and their reference group (hence, changing the reference group changes the slopes).
- The F-statistic and R^2 do not change, because we did not change the total amount of variance between the groups, or the linear relationships among the data.

Other Ways of Creating Dummy Codes

If you do not want to manually code your dummy variables, you can include a categorical variable in a regression model as long as it is saved as a factor. When this is done, the `lm()` function will automatically create dummy codes for you. I usually don't like to do this in my own work, because if I ever decide to change the factor levels at a certain point in my code, if I return to previous portions of the code, it could change my output! This is just an FYI in case you want to do this for yourself.

Because the first level of the factor will always be set as the reference group, you want to make sure what group is set to be the first level.

These lines of code yield the same output:

```
# Model with manually coded dummy variables (Americas as reference group)

coef(lm(infant ~ Africa + Asia, data = Leinhardt3))
```

```
## (Intercept)      Africa        Asia
##    55.12273    87.16845    41.04764
```

```
# Model with factor levels
## First, you need to order the levels (if they're not already ordered the way you
## would like), with the group you want as the reference group listed as the first
## level in the factor

Leinhardt3$region <-
```

```
  factor(Leinhardt3$region,
         levels = c("Americas", "Africa", "Asia"))


## Once the factor levels are ordered, you can use the factor in the regression:
coef(lm(infant ~ region, data = Leinhardt3))
```

```
##  (Intercept) regionAfrica   regionAsia
##     55.12273     87.16845     41.04764
```

There are also functions from other packages that you can use that will add the dummy-coded variables to your data. We will not cover those packages in this lab, but if you'd like to learn more, you can read about the `fastDummies` package here.

Example with 4+ levels

Now that we have done an example of dummy codes with a 3 level grouping variable, lets review an example with a 4 level grouping variable.

We will use the full Leinhardt data, and include *Europe* as a region in the analysis. We will use *Americas* as the reference region again.

```
# Create Dummy Codes
Leinhardt$Africa <- ifelse(Leinhardt$region == "Africa", 1, 0)
Leinhardt$Asia <- ifelse(Leinhardt$region == "Asia", 1, 0)
Leinhardt$Europe <- ifelse(Leinhardt$region == "Europe", 1, 0)

# Summary of Dummy Codes (for demonstration)
Leinhardt %>%
  group_by(region) %>%
  summarise(Africa = unique(Africa),
            Asia = unique(Asia),
            Europe = unique(Europe)) %>%
  flextable()
```

region	Africa	Asia	Europe
Africa	1	0	0
Americas	0	0	0
Asia	0	1	0
Europe	0	0	1

```
# The model
m4 <- lm(infant ~ Africa + Asia + Europe, data = Leinhardt)
summary(m4)
```

```
##
## Call:
## lm(formula = infant ~ Africa + Asia + Europe, data = Leinhardt)
```

```
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -37.52  -5.76  16.91 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    55.12      16.96   3.250 0.001585 **
## Africa         87.17      21.76   4.005 0.000122 ***
## Asia           41.05      22.85   1.797 0.075495 .
## Europe        -35.87      25.28  -1.419 0.159176
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 79.54 on 97 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.2556, Adjusted R-squared:  0.2326
## F-statistic:  11.1 on 3 and 97 DF,  p-value: 2.494e-06
```

$$\widehat{\texttt{infant}}_i = 55.12 + 87.17\texttt{africa}_i + 41.05\texttt{asia}_i - 35.87\texttt{europe}_i$$

```
bind_rows(three_groups = tidy(m3),
          four_groups = tidy(m4),
          .id = "Model") %>%
  flextable() %>%
  colformat_double(digits = 2) %>%
  merge_v(j = 1) %>%
  fix_border_issues() %>%
  theme_vanilla() %>%
  autofit()
```

Model	term	estimate	std.error	statistic	p.value
three_groups	(Intercept)	55.12	18.65	2.96	0.00
	Africa	87.17	23.93	3.64	0.00
	Asia	41.05	25.12	1.63	0.11
four_groups	(Intercept)	55.12	16.96	3.25	0.00
	Africa	87.17	21.76	4.01	0.00
	Asia	41.05	22.85	1.80	0.08
	Europe	-35.87	25.28	-1.42	0.16

```
bind_rows(three_groups = glance(m3),
          four_groups = glance(m4),
          .id = "Model")  %>%
  flextable() %>%
  colformat_double(digits = 2) %>%
  autofit()
```

Model	r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	
three__groups	0.15	0.12	87.46	6.80	0.00	2.00	-487.35	98
four__groups	0.26	0.23	79.54	11.10	0.00	3.00	-583.28	1,17

- The intercept and dummy code variables for Africa and Asia did not change, because they only represent group means. Adding another group does not change the means of the other groups.
- The DF and R^2 changed because we added more another parameter

Interactions with Dummy Codes (Two-Way Factorial ANOVA)

We can also use Dummy Codes to do two-way factorial ANOVA (with interactions), by including a second dummy-coded predictor. We will return to the analysis with only three groups (Americas, Africa, and Asia).

In our model, let's return to using Americas as the reference group, but now also consider whether the country is exports oil. To do this, we can just create a dummy code for whether the country does or does not export oil. We will set "no" as the reference group.

```
Leinhardt3$oil_y = ifelse(Leinhardt3$oil == "yes", 1, 0)
```

Normally when we do interactions in R, we have written them using multiplication. For example, the model we want to test is:

$$\text{Mortality} \quad \text{Region} * \text{Oil Production}$$

In a classical ANOVA, we could enter the variables into the analysis using this same framework as above. However, now that we are using dummy codes, we need to do a little bit of extra work, as the model now looks like:

$$\text{Mortality} \quad (\text{Africa} + \text{Asia}) * \text{Oil Production}$$

We could factor the equation out as follows:

$$\text{Mortality} \quad \text{Africa} + \text{Asia} + \text{Oil} + (\text{Africa} * \text{Oil}) + (\text{Asia} * \text{Oil})$$

We already have the dummy codes for Africa, Asia, and Oil in the data frame. The next step will be for us to manually code the interaction terms for the model.

```
Leinhardt3$Africa.oil_y <- Leinhardt3$Africa * Leinhardt3$oil_y
Leinhardt3$Asia.oil_y <- Leinhardt3$Asia * Leinhardt3$oil_y
```

Here is a summary of the coding:

```
Leinhardt3 %>%
  mutate(oil = as.character(oil)) %>%
  group_by(region, oil) %>%
  summarise(Africa = unique(Africa),
            Asia = unique(Asia),
            Oil = unique(oil_y),
            Oil_x_Africa = unique(Africa.oil_y),
            Oil_x_Asia = unique(Asia.oil_y),
            .groups = "drop") %>%
  flextable() %>%
  theme_zebra() %>%
  autofit()
```

region	oil	Africa	Asia	Oil	Oil_x_Africa	Oil_x_Asia
Americas	no	0	0	0	0	0
Americas	yes	0	0	1	0	0
Africa	no	1	0	0	0	0
Africa	yes	1	0	1	1	0
Asia	no	0	1	0	0	0
Asia	yes	0	1	1	0	1

```
m5 <-
  summary(lm(infant ~ Africa + Asia + oil_y + Africa.oil_y + Asia.oil_y,
           data = Leinhardt3))

m5
```

```
##
## Call:
## lm(formula = infant ~ Africa + Asia + oil_y + Africa.oil_y +
##     Asia.oil_y, data = Leinhardt3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -239.60  -38.73   -8.12   19.52  382.30
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    54.125     18.238   2.968 0.003996 **
## Africa         87.604     23.393   3.745 0.000346 ***
## Asia           20.604     24.694   0.834 0.406658
## oil_y          10.975     60.489   0.181 0.856500
## Africa.oil_y   -4.604     78.045  -0.059 0.953111
## Asia.oil_y    181.996     78.445   2.320 0.022989 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 81.56 on 77 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.2845, Adjusted R-squared:  0.238
## F-statistic: 6.124 on 5 and 77 DF,  p-value: 7.934e-05
```

Interpretations:

- The intercept refers to the expected infant mortality when *Africa*, *Asia*, *oil_y*, *Africa.oil_y*, and *Asia.oil_y* are all equal to 0. Therefore, the intercept here refers to mean infant mortality rate for non-oil-exporting regions in the Americas (this is the reference group). So the average infant mortality rate for non-oil-exporting countries in the Americas is 54.13 per 1,000 live births.

  - Tip: Sometimes it can be tricky to interpret the intercept when you have dummy-coded interactions in the data. Remember, you can ALWAYS apply the phrase "the intercept is the predicted value of y when everything else in the model is 0". Then, all you need to do is figure out what it means for "everything in the model to be 0". If you only have categorical variables in the model, the intercept will ALWAYS be the mean of the sub-group that overlaps both variables' reference group. In other words, *Americas* was the reference group of *region*, and *non-oil producing* was the reference group of *oil_y*. So, the reference group is non-oil-exporting countries in the Americas.

- *Africa* is the difference between African non-oil-exporting regions and American non-oil-exporting regions (the reference group). So the average infant mortality rate for African non-oil exporting regions is $54.13 + 87.60 = 141.73$ per 1,000 live births.
  - To interpret main effects, you will always work at the level of the reference group for the other main effect. So, for the main effects of Africa and Asia, you will always be talking about non-oil exporting countries.
- *Asia* is the difference between Asian non-oil exporting regions and American non-oil-exporting regions. So the average infant mortality rate for Asian non-oil-exporting regions is $54.13 + 20.60 = 74.73$ per 1,000 live births.
- *oil_y* is the difference between American oil-exporting regions and American non-oil-exporting regions. So the average infant mortality rate for American oil-exporting regions is $54.13 + 10.98 = 65.11$ per 1,000 live births.
- *Africa.oil_y* is the amount by which the mean difference of African oil-exporting regions versus American non-oil exporting regions exceeds the simple effects of *Africa* and *oil_y*. So the average infant mortality rate of African oil-exporting regions is $54.13 + 87.60 + 10.98 - 4.60 = 149.11$ per 1,000 live births.
- *Asia.oil_y* is the amount by which the mean difference of Asian oil-exporting regions versus American non-oil exporting regions exceeds the simple effects of *Asia* and *oil_y*. So the average infant mortality rate of Asian oil-exporting regions is $55.13 + 20.60 + 10.98 + 182 = 268.71$ per 1,000 live births.

Here is an example of how we could calculate the predicted mean for Asian Oil-exporting countries. In this example, Asia = 1, Oil = 1, and Asia.Oil = 1; Africa = 0 and Africa.Oil = 0.

$$\widehat{\texttt{infant}}_i = 54.13 + (87.60 * 0) + (20.60 * 1) + (10.98 * 1) - (4.60 * 0) + (182 * 1)$$

This simplifies to:

$$54.13 + 20.60 + 10.98 + 182$$

And equals:

```r
54.13 + 20.60 + 10.98 + 182.00
```

```
## [1] 267.71
```

Again, we are essentially predicting means, so we can verify the above by computing the observed mean

```r
Leinhardt3 %>%
  group_by(region, oil) %>%
  summarise(mean_infant_mortality = mean(infant, na.rm = T))
```

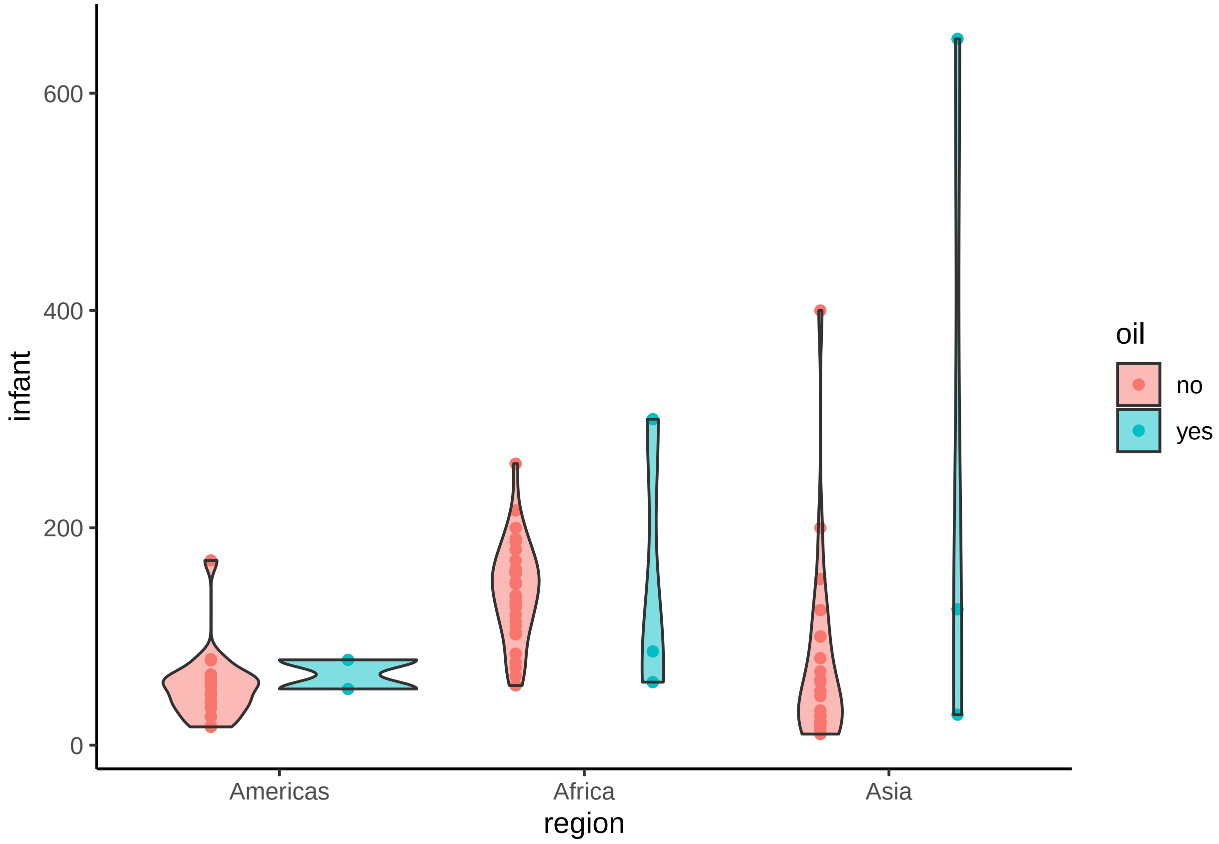
```
## # A tibble: 6 x 3
## # Groups:   region [3]
##   region   oil   mean_infant_mortality
##   <fct>    <fct>                 <dbl>
## 1 Americas no                     54.1
## 2 Americas yes                    65.1
## 3 Africa   no                    142.
## 4 Africa   yes                   148.
## 5 Asia     no                     74.7
## 6 Asia     yes                   268.
```

Here are some graphs that illustrate the regression we just did.

```
ggplot(data = Leinhardt3,
       aes(x = region, y = infant)) +
  geom_jitter(aes(col = oil),
              alpha = 1,
              position = position_dodge(width = 0.9)) +
  geom_violin(aes(fill = oil),
              alpha = 0.5) +
  theme_classic()
```

```
## Warning: Removed 4 rows containing non-finite values (‘stat_ydensity()‘).
```

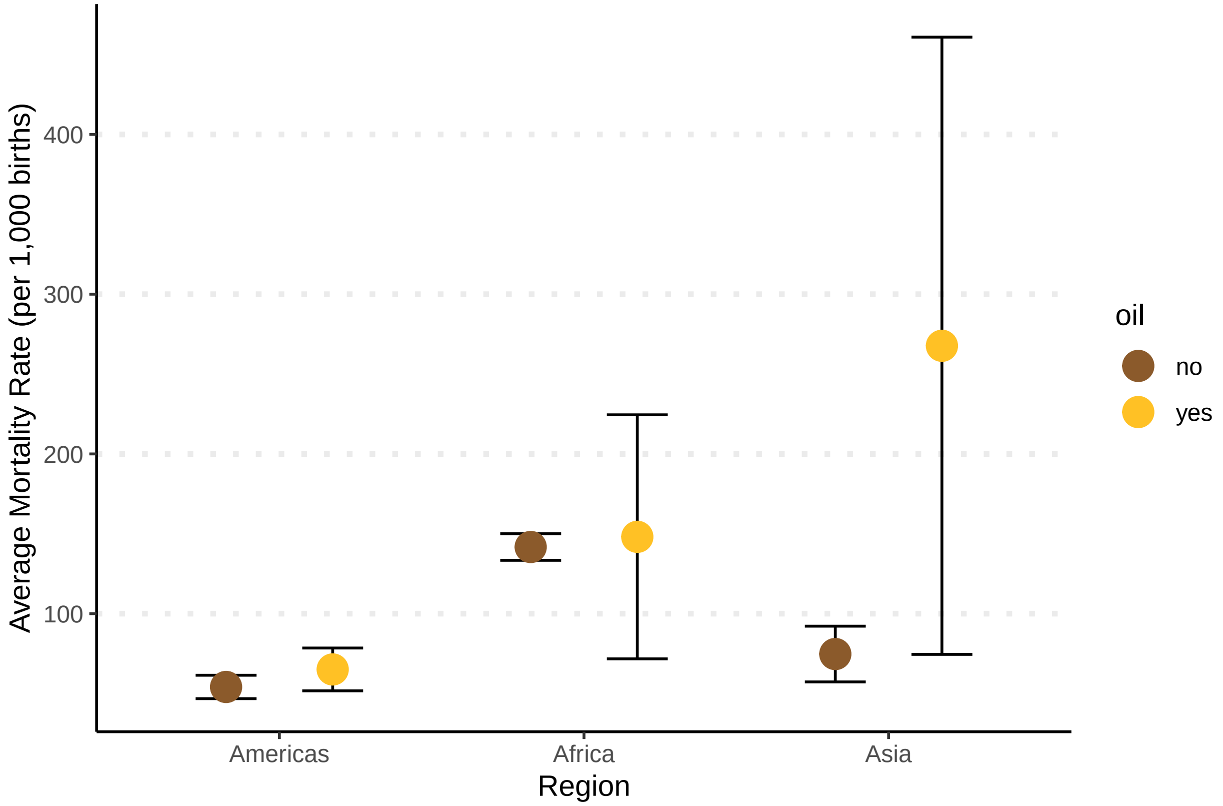
```
## Warning: Removed 4 rows containing missing values (‘geom_point()‘).
```



```
leinhardt_means2 <-
  Leinhardt3 %>%
  filter(!is.na(infant)) %>%
  group_by(region, oil) %>%
  summarise(mean = mean(infant, na.rm = T),
            n = n(),
            sd = sd(infant),
            se = sd / sqrt(n),
            ub = mean + se,
            lb = mean - se)
```

```
## ‘summarise()‘ has grouped output by ’region’. You can override using the
## ‘.groups‘ argument.
```

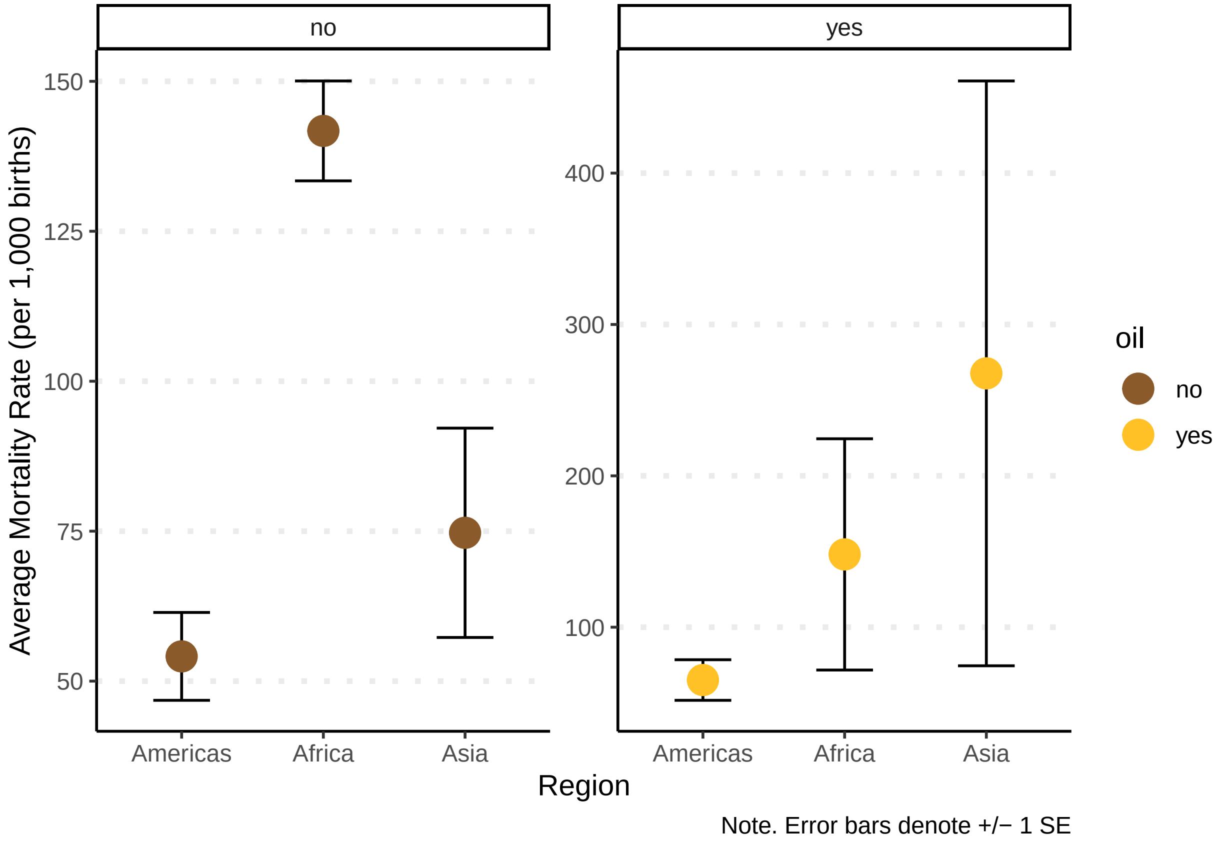
```
ggplot(data = leinhardt_means2,
       aes(x = region, y = mean)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, group = oil),
                width = 0.4,
                position = position_dodge(width = 0.7)) +
  geom_point(aes(col = oil),
             size = 5,
             position = position_dodge(width = 0.7)) +
  scale_color_manual(values = c("tan4", "goldenrod1")) +
  labs(x = "Region",
       y = "Average Mortality Rate (per 1,000 births)",
       caption = "Note. Error bars denote +/- 1 SE") +
  theme_classic() +
  theme(panel.grid.major.y = element_line(linetype = "dotted", size = 1))
```



Note. Error bars denote +/− 1 SE

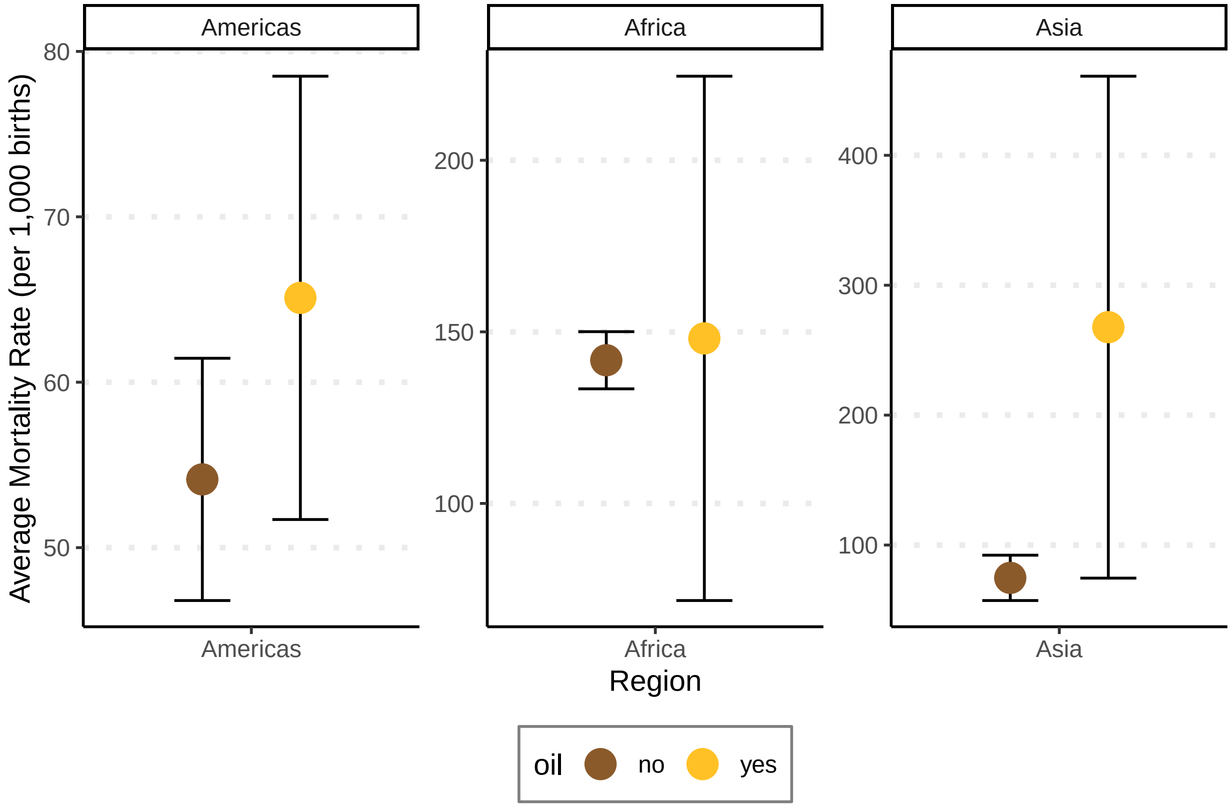
We could also facet our plot to focus on specific comparisons, e.g., averages mortality rates between regions depending on whether or not they export oil

```
ggplot(data = leinhardt_means2,
       aes(x = region, y = mean)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, group = oil),
                width = 0.4,
                position = position_dodge(width = 0.7)) +
  geom_point(aes(col = oil),
             size = 5,
             position = position_dodge(width = 0.7)) +
  scale_color_manual(values = c("tan4", "goldenrod1"))+
  labs(x = "Region",
       y = "Average Mortality Rate (per 1,000 births)",
       caption = "Note. Error bars denote +/- 1 SE") +
  theme_classic() +
  theme(panel.grid.major.y = element_line(linetype = "dotted", size = 1)) +
  facet_wrap(~ oil, scales = "free")
```



or averages mortality rates between oil exporting and non-exporting regions depending geographical region

```
ggplot(data = leinhardt_means2,
       aes(x = region, y = mean)) +
  geom_errorbar(aes(ymin = lb, ymax = ub, group = oil),
                width = 0.4,
                position = position_dodge(width = 0.7)) +
  geom_point(aes(col = oil),
             size = 5,
             position = position_dodge(width = 0.7)) +
  scale_color_manual(values = c("tan4", "goldenrod1"))+
  labs(x = "Region",
       y = "Average Mortality Rate (per 1,000 births)",
       caption = "Note. Error bars denote +/- 1 SE") +
  theme_classic() +
  theme(panel.grid.major.y = element_line(linetype = "dotted", size = 1),
        legend.position = "bottom",
        legend.background = element_rect(color = "grey50")) +
  facet_wrap(~ region, scales = "free")
```



Note. Error bars denote +/− 1 SE

Notice how different comparisons stand out depending on how you graph the data!

Comments on Significance:

Lets return again to the model and interpret what is and is not significant. For reference, here is the regression:

```
m5
```

```
##
## Call:
## lm(formula = infant ~ Africa + Asia + oil_y + Africa.oil_y +
##     Asia.oil_y, data = Leinhardt3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -239.60  -38.73   -8.12   19.52  382.30
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    54.125     18.238   2.968 0.003996 **
## Africa         87.604     23.393   3.745 0.000346 ***
## Asia           20.604     24.694   0.834 0.406658
## oil_y          10.975     60.489   0.181 0.856500
## Africa.oil_y   -4.604     78.045  -0.059 0.953111
## Asia.oil_y    181.996     78.445   2.320 0.022989 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 81.56 on 77 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.2845, Adjusted R-squared:  0.238
## F-statistic: 6.124 on 5 and 77 DF,  p-value: 7.934e-05
```

- Overall, the model is significant ($F(5, 77) = 6.12$, $p < .001$), suggesting that *region*, *oil*, and/or the interaction between these variables has an effect on *infant* mortality.
- The average of non-oil exporting countries in the Americas (i.e., the reference group, represented by the *intercept*) is significantly higher than 0 ($M = 54.13$, p = .004).
- The mean of non-oil exporting countries in *Africa* is significantly higher ($M = 141.73$, $p < .001$) than the mean of non-oil exporting countries in the Americas ($M = 54.13$; i.e., the reference group).
- The mean of non-oil exporting countries in *Asia* ($M = 74.73$) is not significantly different ($p = .41$) than the mean of non-oil exporting countries in the Americas ($M = 54.13$; i.e., the reference group).
- The mean of *oil* exporting countries in the Americas ($M = 65.11$) is not significantly different ($p = .86$) than the mean of non-oil exporting countries in the Americas ($M = 54.13$; i.e., the reference group).
- The mean of *oil* exporting countries in *Africa* ($M = 148.12$) is not significantly different ($p = .95$) from the mean of non-oil exporting countries in the Americas ($M = 54.13$; i.e., the reference group).
- The mean of *oil* exporting countries in *Asia* ($M = 267.71$) is significantly higher ($p = .023$) from the mean of non-oil exporting countries in the Americas ($M = 54.13$; i.e., the reference group).

Effect Coding

An alternative to dummy coding is to use effect coding. Effect coding allows us to make a variety of different comparisons. For example, comparing the mean of each group to the grand mean. However, because we can only make $J - 1$ comparisons, we have to drop a level in our effect coding.

One-way ANOVA

In effect coding, the reference group (the group whose estimate will be dropped) is coded as -1, the group represented by the effect code is coded as 1, and all other groups are coded as 0.

Let's look at how we can use effect coding to conduct the one-way ANOVA using region to predict infant morality rate.

Let's create contrast-coded dummy variables, using *Americas* as our reference:

- *Africa_E* = -1 if region = Americas
- *Asia_E* = -1 if region = Americas
- *Africa_E* = 1 if region = Africa
- *Africa_E* = 0 if region = Asia
- *Asia_E* = 0 if region = Africa
- *Asia_E* = 1 if region = Asia

This chunk creates this effect codes, and provides a summary table.

```
Leinhardt3$Africa_E <- dplyr::recode(Leinhardt3$region,
                                     Americas = -1,
                                     Africa = 1,
                                     Asia = 0)

Leinhardt3$Asia_E <- dplyr::recode(Leinhardt3$region,
                                   Americas = -1,
                                   Africa = 0,
                                   Asia = 1)

Leinhardt3 %>% group_by(region) %>%
  summarise(AfricaE = unique(Africa_E),
            AsiaE = unique(Asia_E))
```

```
## # A tibble: 3 x 3
##   region   AfricaE AsiaE
##   <fct>      <dbl> <dbl>
## 1 Americas      -1    -1
## 2 Africa         1     0
## 3 Asia           0     1
```

```
#or, we could instead use base R
Leinhardt$Africa_E <- c('Americas'=-1,'Africa'=1,'Asia'=0)
Leinhardt$Asia_E <- c('Americas'=-1,'Africa'=0,'Asia'=1)
```

Now, using our effect-coded variables to predict infant mortality:

```
m6 <- lm(infant ~ Africa_E + Asia_E, data = Leinhardt3)
summary(m6)
```

```
##
## Call:
## lm(formula = infant ~ Africa_E + Asia_E, data = Leinhardt3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -43.23  -9.12  18.96 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   97.861      9.752  10.035 8.26e-16 ***
## Africa_E      44.430     13.042   3.407  0.00103 **
## Asia_E        -1.691     13.767  -0.123  0.90255
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 87.46 on 80 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.124
## F-statistic: 6.803 on 2 and 80 DF,  p-value: 0.001869
```

Alternatively, we can use R to create the contrasts for us. Here, the last level of our factor is the one that gets dropped

```r
Leinhardt3$region2 <- factor(Leinhardt3$region,
                             levels = c("Africa", "Asia", "Americas"))

contrasts(Leinhardt3$region2) <- contr.sum(3)

summary(lm(infant ~ region2, data = Leinhardt3))
```

```
##
## Call:
## lm(formula = infant ~ region2, data = Leinhardt3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -43.23  -9.12  18.96 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   97.861      9.752  10.035 8.26e-16 ***
## region21      44.430     13.042   3.407  0.00103 **
## region22      -1.691     13.767  -0.123  0.90255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87.46 on 80 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.124
## F-statistic: 6.803 on 2 and 80 DF,  p-value: 0.001869
```

$$\widehat{\texttt{infant}}_i = 97.86 + 44.43\texttt{AfricaE}_i - 1.69\texttt{AsiaE}_i$$

Interpretation:

- The *intercept* is the mean of the means. If the design is balanced (the same number of subjects or rows in each group), the mean of means will be the same thing as the grand mean. Since this analysis is not balanced, these are not the same.

```r
# grand mean
mean(Leinhardt3$infant, na.rm = T)
```

```
## [1] 104.1831
```

```r
# Mean of means
Leinhardt3 %>%
  group_by(region) %>%
  summarise(mean_mortality_rate = mean(infant, na.rm = T)) %>%
  summarise(mean_of_means = mean(mean_mortality_rate, na.rm = T)) %>%
  unlist()
```

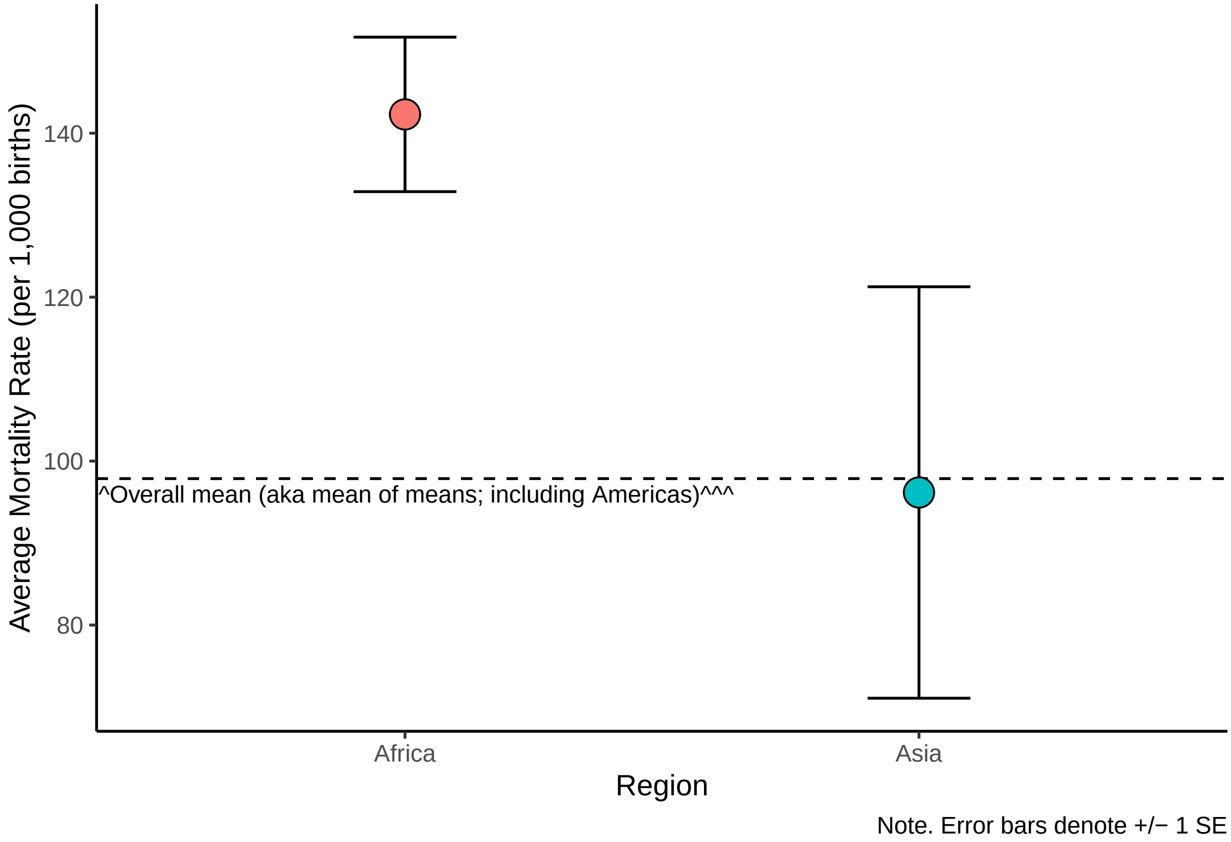
```
## mean_of_means
##      97.86142
```

Now, continuing with the interpretation (note that we will call the mean of means the “overall mean”)

- *Africa_E* is the amount by which Africa differs from the overall mean. So the mean African infant mortality rate is 142.29 per 1,000 live births.
- *Asia_E* is the amount by which Asia differs from the overall mean. So the mean Asian infant mortality rate is 96.17 per 1,000 live births.

Here is a visual summary of the analysis we did.

```
leinhardt_means %>%
  filter(region != "Americas") %>%
  ggplot(aes(x = region, y = mean, fill = region)) +
  geom_hline(yintercept = coef(m6)[1], linetype = "dashed") +
  geom_errorbar(aes(ymin = lb, ymax = ub),
                width = 0.2) +
  geom_point(size = 5, shape = 21) +
  guides(fill = "none") +
  annotate(geom = "text",
           label = "^^^Overall mean (aka mean of means; including Americas)^^^",
           x = 1, y = 96, size = 3) +
  labs(x = "Region",
       y = "Average Mortality Rate (per 1,000 births)",
       caption = "Note. Error bars denote +/- 1 SE") +
  theme_classic()
```



Note, there is no estimate for the reference group (Americas), as we have run out of DF to make an orthogonal estimate for this group (because we have estimated the mean of means instead; the intercept).

Comment on Significance:

Lets return to the model and interpret what is and is not significant. For reference, here is the output:

```
summary(m6)
```

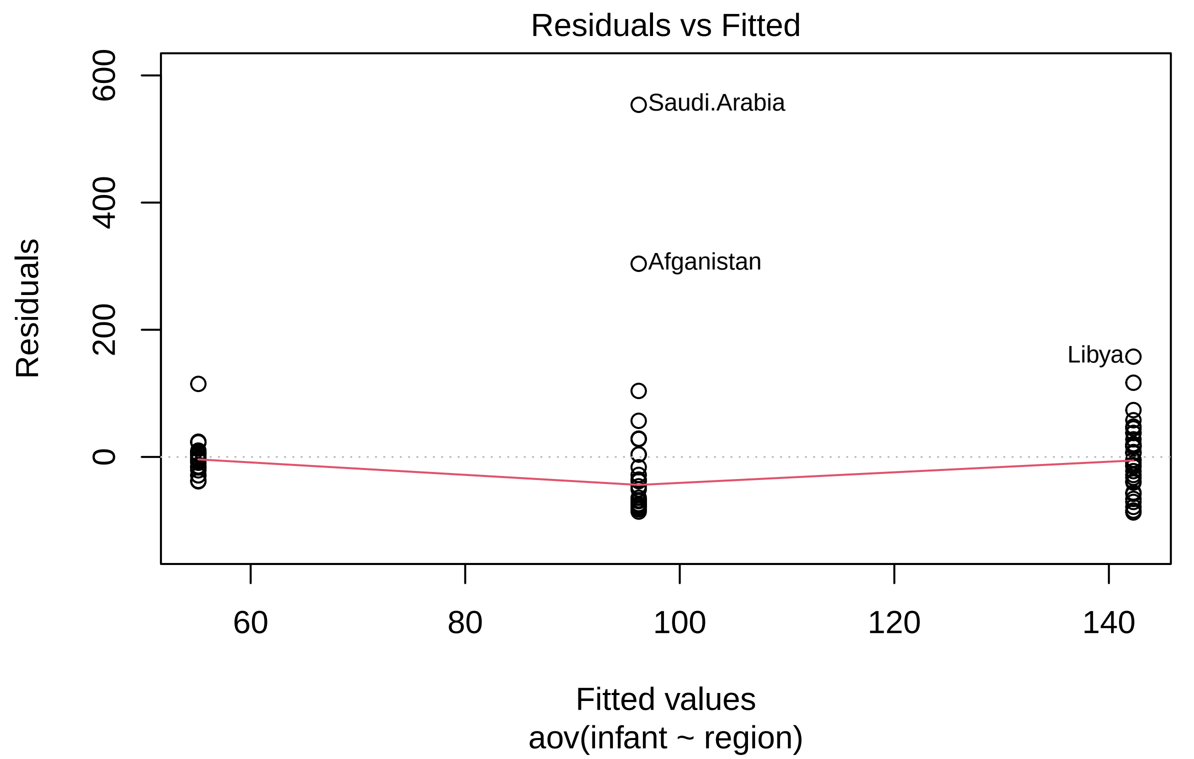
```
##
## Call:
## lm(formula = infant ~ Africa_E + Asia_E, data = Leinhardt3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -87.29 -43.23  -9.12  18.96 553.83
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   97.861      9.752  10.035 8.26e-16 ***
## Africa_E      44.430     13.042   3.407  0.00103 **
## Asia_E        -1.691     13.767  -0.123  0.90255
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87.46 on 80 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.1453, Adjusted R-squared:  0.124
## F-statistic: 6.803 on 2 and 80 DF,  p-value: 0.001869
```

- Overall, the model is significant (F(2,80) = 6.80, p = .002), suggesting that *region* has an effect on infant mortality, and that at least one group differs from the overall mean.
- The overall mean (M = 97.86, represented by the *intercept*) is significantly greater than 0 (p < .001). (Note: This may or may not be interesting or relevant to the hypotheses you are testing; in our case it is not that interesting, since it is not possible to have negative infant mortality)
- The mean infant mortality of countries in *Africa* (M = 142.29) is significantly higher (p = .001) than the overall mean across the three regions (M = 97.86).
- The mean infant mortality of countries in *Asia* (M = 96.17) is not significantly different (p = .90) from the overall mean across the three regions (M = 97.86).

If you would like to know more about effects codes, check out this link. For examples of how to do effect coded regressions in R, click here. There is information about a lot of different types of coding schemes here.

#Small section on testing the assumptions of ANOVA

```
resid.aov <- aov(infant ~ region, data = Leinhardt3)
#Leinhardt3
plot(resid.aov,1) #residuals vs. fits plot. This can show outliers in our data that can affect normalit
```



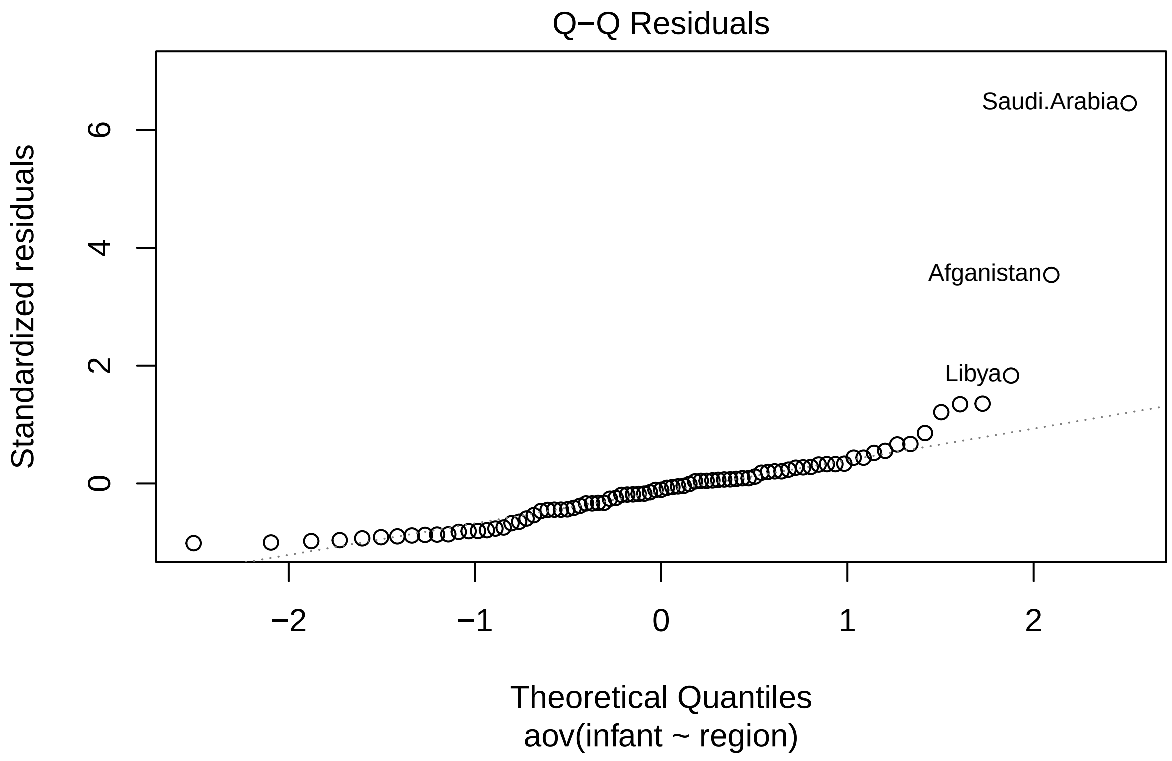
```
leveneTest(infant ~ region, data=Leinhardt3) #Levene's test, if significant, shows that variance across
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value  Pr(>F)
## group  2  2.8944 0.06115 .
##       80
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#If Levene's is significant, we can still use account for this with certain methods or approaches to th


#Checking normailty can utilize Q-Q plots, boxplots to visualize the data, the shapiro wilk's test.

plot(resid.aov,2)
```



```
aov_resids_vals <- residuals(object=resid.aov)
shapiro.test(aov_resids_vals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  aov_resids_vals
## W = 0.66586, p-value = 1.898e-12
```

```
#if this is violated, the F test is still somewhat robust to nonnormality but only in samples that are
```