General Linear Model

So far

- Reviewed last semester and previewed this semester
  - Theme 1: The Backstreet Boys
  - Theme 2: What is 0?
- Put some checks and balances in place to minimize *procedural* errors as much as possible

Thinking in terms of models

- All the tests run thus far can be thought of as a model for how you think the world works
- Our DV (here forth Y) is what we are trying to understand
- We hypothesize it has some relationship with your IV(s) (here forth Xs), with what is left over described as error (E)
- $Y = X + E$

See this in our R code

Independent samples t-test

```
t.1 <- t.test(y ~ x, data = d)
# y is cont and x is a categoriocal/nominal (dichotomous) fac
```

One-way ANOVA

```
a.1 <- aov(y ~ x, data=d)
# y is cont and x is a categoriocal/nominal factor
```

General linear model (GLM)

- This model (equation) can be very simple as in a treatment/control experiment
- It can be very complex in terms of trying to understand something like academic achievement
- The majority of our models fall under the umbrella of a general(ized) linear model
- Models imply our theory about how the data are generated (ie how the world works)

$$Y_i = b_0 + b_1 X_i + e_i$$

- Y / DV / Outcome / Response / Criterion
- X / IV / Predictor / Explanatory variable
- Regression coefficient (weight) / b / $b*$ / β
- Intercept b_0 / β_0
- Error / Residuals e
- Predictions $\hat{Y}$

- $Y_i \sim Normal(\mu, \sigma)$
- The DV, Y for each person i is distributed normally, with a mean of μ and a standard deviation of σ

Regression models

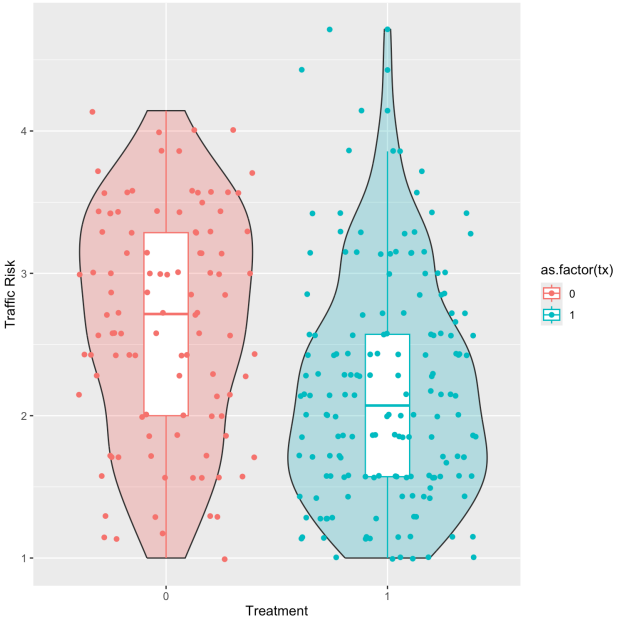
- These models are a way to convey the relationship between two (or more) variables. They translate our hypotheses into math.
- We can use these models to get information we may be interested in (e.g. means, SEs) and test hypotheses about the relationship among variables
- *"All models are wrong but some are useful (and some are better than others)"* - George Box

Example data can be found on GitHub, or here: exampleData.csv

```
example.data
```

```
## # A tibble: 270 × 3
##       id    tx traffic.risk
##    <dbl> <dbl>        <dbl>
##  1     1     1         1.86
##  2     2     1         1
##  3     3     1         3.29
##  4     4     1         2
##  5     5     1         2.43
##  6     6     1         3.29
##  7     7     0         1.17
##  8     8     0         2.43
##  9     9     0         3
## 10    10     1         1.71
## # ℹ 260 more rows
```

```
ggplot(example.data,
       aes(x = as.factor(tx)
           y = traffic.risk)
  geom_violin(aes(fill = as.
              alpha = .3,
              show.legend =
  geom_boxplot(aes(color = a
               fill = "white
               width = .2) +
  geom_jitter(aes(color = as
  labs(x = "Treatment",
       y = "Traffic Risk")
```



```
t.test(traffic.risk ~ tx, data = example.data,
               var.equal = TRUE)
```

```
##
##     Two Sample t-test
##
## data:  traffic.risk by tx
## t = 4.9394, df = 268, p-value = 0.000001381
## alternative hypothesis: true difference in means between group 0 a
## 95 percent confidence interval:
##  0.2893360 0.6728755
## sample estimates:
## mean in group 0 mean in group 1
##        2.650641        2.169535
```

```
a.1 <- aov(traffic.risk ~ tx, data = example.data)
summary(a.1)
```

```
##               Df Sum Sq Mean Sq F value     Pr(>F)
## tx             1   14.8  14.800    24.4 0.00000138 ***
## Residuals    268  162.6   0.607
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod.1 <- lm(traffic.risk ~ tx, data = example.data)
summary(mod.1)
```

```
##
## Call:
## lm(formula = traffic.risk ~ tx, data = example.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65064 -0.59811 -0.02668  0.54475  2.54475
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  2.65064    0.07637  34.707 < 0.0000000000000002 ***
## tx          -0.48111    0.09740  -4.939              0.00000138 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7789 on 268 degrees of freedom
## Multiple R-squared:  0.08344,    Adjusted R-squared:  0.08002
## F-statistic:  24.4 on 1 and 268 DF,  p-value: 0.000001381
```

```
mod.1 <- lm(traffic.risk ~ tx, data = example.data)
anova(mod.1)
```

```
## Analysis of Variance Table
##
## Response: traffic.risk
##            Df Sum Sq Mean Sq F value      Pr(>F)
## tx          1  14.80 14.7999  24.398 0.000001381 ***
## Residuals 268 162.57  0.6066
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example summary

Same *p*-values for each test; same SS; same test!

- *t*-test gives you a *t* & df (output may also give you group mean and SD)
- ANOVA gives you an SSs, dfs, MS and *F*s
- Linear model (regression) gives you an equation (and SSs and *F*s)
- Which one is more useful?

ANOVA as regression

$$Y_i = b_0 + b_1 X_i + e_i$$

$$T.risk_i = b_0 + b_1 TX_i + e_i$$

- Each individual has a unique Y value an X value and a residual/error term
- The model only has a single b_0 and b_1 term. These are the regression parameters. b_0 is the intercept and b_1 quantifies the relationship between your model of the world and the DV.

What do the estimates tell us?

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    2.65     0.0764     34.7  3.80e-101
## 2 tx            -0.481    0.0974     -4.94 1.38e-  6
```

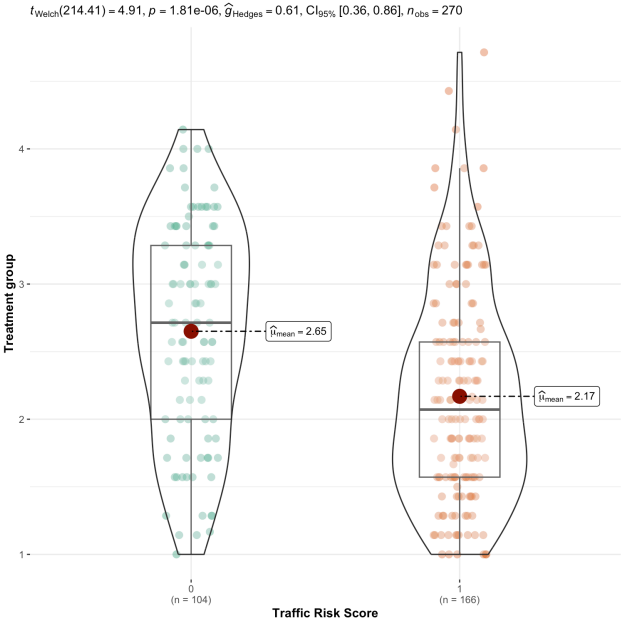
```
example.data %>%
  group_by(tx) %>%
  summarise(mean(traffic.risk))
```

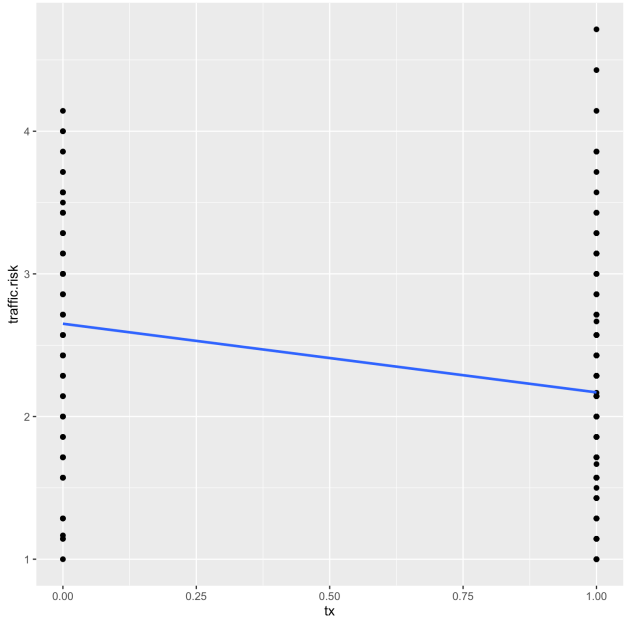
```
## # A tibble: 2 × 2
##      tx `mean(traffic.risk)`
##   <dbl>                <dbl>
## 1     0                 2.65
## 2     1                 2.17
```

How to interpret regression estimates

- Intercept is the mean of group of variable tx that is coded 0
- Regression coefficient is the slope or rise over run, scaled as a 1 unit on the x axis
- **"For a one unit change in X, there is a b1 predicted change in Y."**
- Regression coefficient is the difference in means between the groups, given that we coded our groups as 0 and 1.

```r
library(ggstatsplot)
ggstatsplot::ggbetweenstats(
  data = example.data,
  x = tx,
  y = traffic.risk,
  xlab = "Traffic Risk Score",
  ylab = "Treatment group",
  bf.message = FALSE,
  messages = FALSE
)
```





This is what it looks like if we wanted to put a "regression line" to the data. Note that the same interpretation for a regression line holds: for a 1 unit change in X (tx) there is a predicted b change in Y (traffic risk)

Interpretations

- Intercept (b_0) signifies the level of Y when your model IVs (Xs) are zero
- Regression (b_1) signifies the difference for a one unit change in your X
- As with last semester you have estimates (like $\bar{x}$) and standard errors, which you can then ask whether they are likely assuming a null or create a CI

Predicted scores

- Based on the output how do I calculate means for each group?

```
tidy(mod.1)
```

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    2.65     0.0764     34.7  3.80e-101
## 2 tx            -0.481    0.0974     -4.94 1.38e-  6
```

ANOVA as regression

- "For a one unit change in X, there is a b_1 predicted change in Y" -- always true
- Nominal/categorical variables do not have any inherent numbers associated with them so we need to assign them numbers
- What numbers you assign will impact the equation/estimates/hypothesis you can test
- Make them useful! O and 1 are useful and are the default in R

ANOVA as regression

$$T.risk_i = b_0 + b_1 TX_i + e_i$$

```
    ## # A tibble: 270 × 3
##       id    tx traffic.risk
##    <dbl> <dbl>        <dbl>
##  1     1     1         1.86
##  2     2     1         1
##  3     3     1         3.29
##  4     4     1         2
##  5     5     1         2.43
##  6     6     1         3.29
##  7     7     0         1.17
##  8     8     0         2.43
##  9     9     0         3
## 10    10     1         1.71
     ## # ℹ 260 more rows
```

```
library(dplyr)
example.data$tx.r <- as.factor(example.data$tx)
example.data$tx.r <- recode_factor(example.data$tx.r, "0" = "
```

Create a new variable that is not numeric

```
example.data
```

```
## # A tibble: 270 × 4
##       id    tx traffic.risk tx.r
##    <dbl> <dbl>        <dbl> <fct>
##  1     1     1         1.86 treatment
##  2     2     1         1    treatment
##  3     3     1         3.29 treatment
##  4     4     1         2    treatment
##  5     5     1         2.43 treatment
##  6     6     1         3.29 treatment
##  7     7     0         1.17 control
##  8     8     0         2.43 control
##  9     9     0         3    control
## 10    10     1         1.71 treatment
## # ℹ 260 more rows
```

```
mod.1 <- lm(traffic.risk ~ tx, data = example.data)
tidy(mod.1)
```

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    2.65     0.0764     34.7  3.80e-101
## 2 tx            -0.481    0.0974     -4.94 1.38e-  6
```

```
mod.1r <- lm(traffic.risk ~ tx.r, data = example.data)
tidy(mod.1r)
```

```
## # A tibble: 2 × 5
##   term          estimate std.error statistic   p.value
##   <chr>            <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)      2.65     0.0764     34.7  3.80e-101
## 2 tx.rtreatment   -0.481    0.0974     -4.94 1.38e-  6
```

What if we changed 0 and 1 to other values?

- Infinite number of ways to code categorical/nominal variables, only a few meaningful ways
- The R default is called "dummy coding"
- Uses 0s and 1s to put numbers to categories. We will soon see what this looks like when you have more than 2 groups.
- Changing the numbers changes...?

Effect coding

```
example.data$tx.effect <- dplyr::recode(example.data$tx,
```

```
example.data
```

```
## # A tibble: 270 × 5
##       id    tx traffic.risk tx.r      tx.effect
##    <dbl> <dbl>        <dbl> <fct>          <dbl>
##  1     1     1         1.86 treatment          1
##  2     2     1         1    treatment          1
##  3     3     1         3.29 treatment          1
##  4     4     1         2    treatment          1
##  5     5     1         2.43 treatment          1
##  6     6     1         3.29 treatment          1
##  7     7     0         1.17 control           -1
##  8     8     0         2.43 control           -1
##  9     9     0         3    control           -1
## 10    10     1         1.71 treatment          1
## # ℹ 260 more rows
```

Effect coding

```
mod.1.eff <- lm(traffic.risk ~ tx.effect, data = example.data
tidy(mod.1.eff)
```

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    2.41     0.0487     49.5  8.15e-137
## 2 tx.effect     -0.241    0.0487     -4.94 1.38e-  6
```

- systematically changes both the intercept and the regression estimate

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    2.41     0.0487     49.5  8.15e-137
## 2 tx.effect     -0.241    0.0487     -4.94 1.38e-  6
```

```
## # A tibble: 2 × 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)    2.65     0.0764     34.7  3.80e-101
## 2 tx            -0.481    0.0974     -4.94 1.38e-  6
```

- Intercept: value when your predictor X is zero. *WHAT DOES ZERO MEAN?!*
- Regression coefficient: one unit increase in X is associated with a (regression estimate) predicted increase in Y

Effect coding

Consists of -1, 1s (And zeros for more than 2 groups)

1. The intercept is the "grand mean" or "mean of means" if unbalanced
2. The regression coefficient represents the group "effect":
    - the difference between the mean of means and the group labeled 1 (we will revisit this when we have more than 2 groups)

Dummy coding

- More appropriate when you are interested in comparing to a specific group rather than an "average person"
- Intercept: value of the group coded zero
- Regression coefficient: mean difference between groups

Contrast coding

- As our models get more complex our coding schemes can too
- What happens if you code the groups -.5, -.5, and 1?
- These make more sense when we have more groups. More groups require more independent variables

Statistical Inference

- The way the world is = our model + error
- How good is our model? Is it a good representation of reality? Does it "fit" the data well?
- Need to go beyond asking if it is significant, because what does that mean?
- We are going to make predictions and see if the predictions (based on our model) matches our data
- We can then compare one model to another to see which one matches our data better. Which one is a better representation of reality?

Predictions

- predictions $\hat{Y}$ are of the form of $E(Y|X)$
- They are created by simply plugging a persons Xs into the created model
- If you have b_0, b_1, and Xs then you can create a prediction

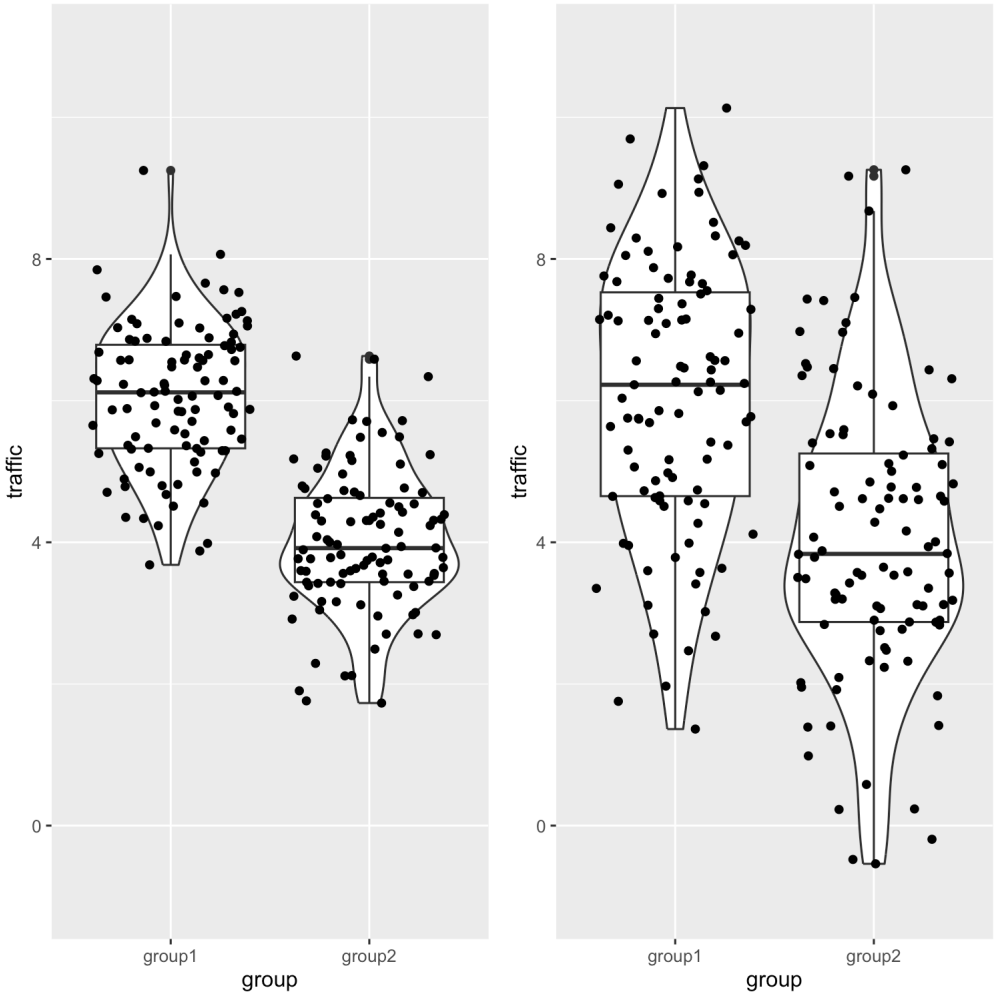
$\hat{Y}_i$ = 2.65064 + -0.48111* X_i

- You have already done this with dummy codes above

Predictions

- We want our predictions to be close to our actual data for each person (Y_i)
- The difference between the actual data and our our prediction ($Y_i - \hat{Y}_i = e$) is the residual, how far we are "off". This tells us how good our fit is.
- You can have the same estimates for two models but completely different fit.
- Previously you have evaluated fit by looking at Eta Squared, SS Error and visualizing observations around group means

Which one has better fit?



Easy to examine fit with `lm` objects

- These are automatically created anytime you run a `lm` in R

```
mod.1 <- lm(traffic.risk ~ tx, data = example.data)
```

```
coefficients(mod.1)        # coefficients
residuals(mod.1)           # residuals
fitted.values(mod.1)       # fitted values ie predicted
summary(mod.1)$r.squared   # R-sq for the model
summary(mod.1)$sigma       # sd of residuals
```

```
coefficients(mod.1)
```

```
## (Intercept)          tx
##   2.6506410  -0.4811057
```

```
fitted.values(mod.1)
```

```
##        1        2        3        4        5        6        7
## 2.169535 2.169535 2.169535 2.169535 2.169535 2.169535 2.650641 2.65064
##        9       10       11       12       13       14       15       1
## 2.650641 2.169535 2.169535 2.650641 2.650641 2.650641 2.650641 2.65064
##       17       18       19       20       21       22       23       2
## 2.650641 2.650641 2.650641 2.169535 2.169535 2.169535 2.169535 2.16953
##       25       26       27       28       29       30       31       3
## 2.650641 2.650641 2.650641 2.650641 2.650641 2.650641 2.650641 2.65064
##       33       34       35       36       37       38       39       4
## 2.650641 2.650641 2.650641 2.650641 2.169535 2.169535 2.169535 2.16953
##       41       42       43       44       45       46       47       4
## 2.650641 2.169535 2.650641 2.650641 2.169535 2.650641 2.650641 2.65064
##       49       50       51       52       53       54       55       5
## 2.169535 2.650641 2.169535 2.169535 2.169535 2.169535 2.169535 2.65064
##       57       58       59       60       61       62       63       6
## 2.169535 2.169535 2.169535 2.169535 2.650641 2.169535 2.169535 2.65064
##       65       66       67       68       69       70       71       7
## 2.169535 2.169535 2.650641 2.650641 2.169535 2.169535 2.169535 2.16953
##       73       74       75       76       77       78       79       8
## 2.169535 2.169535 2.169535 2.169535 2.650641 2.169535 2.169535 2.65064
##       81       82       83       84       85       86       87       8
## 2.169535 2.650641 2.650641 2.650641 2.650641 2.169535 2.169535 2.65064
##       89       90       91       92       93       94       95       9
## 2.169535 2.169535 2.169535 2.650641 2.169535 2.169535 2.169535 2.16953
```

```
residuals(mod.1)
```

```
##            1            2            3            4            5
## -0.312392427 -1.169535284  1.116179002 -0.169535284  0.259036145  1.11
##            7            8            9           10           11
## -1.483974359 -0.222069597  0.349358974 -0.455249570  1.687607573 -0.07
##           13           14           15           16           17
##  0.206501831  0.920787545  1.349358974  0.349358974 -0.936355312 -1.36
##           19           20           21           22           23
## -0.364926740 -0.312392427 -0.598106713 -0.026678141  0.116179002 -0.45
##           25           26           27           28           29
##  1.063644688 -0.650641026 -0.079212455  0.206501831  0.777930403 -1.07
##           31           32           33           34           35
##  0.063644688 -0.507783883  0.777930403 -0.936355312 -1.079212455 -1.07
##           37           38           39           40           41
##  0.116179002 -0.598106713 -0.598106713  0.116179002  0.635073260 -0.59
##           43           44           45           46           47
## -0.507783883  0.635073260  0.116179002 -0.793498169  0.635073260  1.34
##           49           50           51           52           53
##  0.830464716  0.492216117  0.497131383 -0.598106713 -0.312392427 -0.88
##           55           56           57           58           59
##  0.973321859 -0.222069597 -1.169535284  0.259036145 -0.455249570 -0.16
##           61           62           63           64           65
## -0.079212455  0.544750430  1.259036145  0.920787545  0.830464716  2.25
##           67           68           69           70           71
## -0.650641026 -1.650641026 -0.169535284  1.687607573 -1.026678141  0.40
```

Pop quiz

$$\hat{Y}_i = b_0 + b_1 X_i$$

$$Y_i = b_0 + b_1 X_i + e_i$$

$$Y_i - \hat{Y}_i = e$$

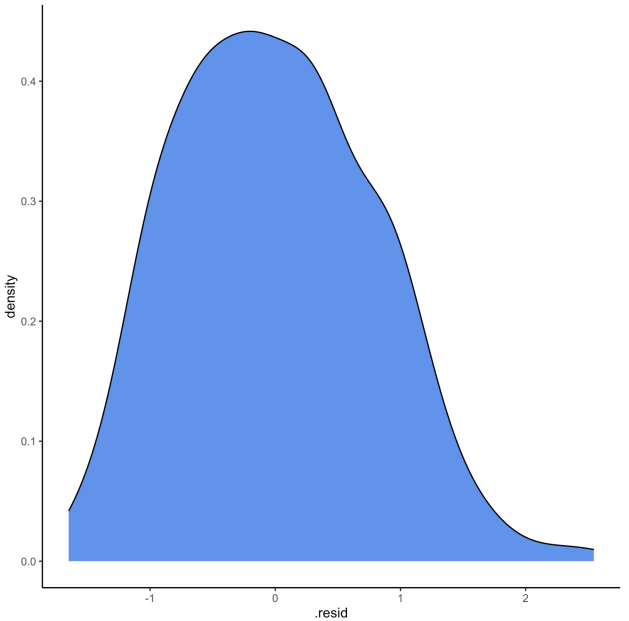
- Can you plug in numbers and calculate subject 3's predicted and residual scores without explicitly asking for lm object residuals and fitted values? (using the same model from slide 37 on)
- Post answers in Slack -- see if you and your peers get the same results!

```
example.data
```

```
## # A tibble: 270 × 5
##       id    tx traffic.risk tx.r      tx.effect
##    <dbl> <dbl>        <dbl> <fct>          <dbl>
##  1     1     1         1.86 treatment          1
##  2     2     1         1    treatment          1
##  3     3     1         3.29 treatment          1
##  4     4     1         2    treatment          1
##  5     5     1         2.43 treatment          1
##  6     6     1         3.29 treatment          1
##  7     7     0         1.17 control           -1
##  8     8     0         2.43 control           -1
##  9     9     0         3    control           -1
## 10    10     1         1.71 treatment          1
## # ℹ 260 more rows
```

Residuals

```
ggplot(fit.1.data,
       aes(.resid)) +
    geom_density(fill = "cc
  theme_classic()
```



`lm` objects

- `lm` objects consist of the information embedded in your linear model
- the `broom` package makes model objects into dataframes

```
library(broom)
fit.1.tidy <- tidy(mod.1)
fit.1.tidy
```

```
## # A tibble: 2 × 5
##   term        estimate std.error stat
##   <chr>          <dbl>     <dbl>
## 1 (Intercept)    2.65     0.0764
## 2 tx            -0.481    0.0974
```

- `Augment` function from the `broom` package amends the original dataset with lm object content. The new variable names of have a "." in front of the name to distinguish

```
fit.1.data <- augment(mod.1)
head(fit.1.data)
```

```
## # A tibble: 6 × 8
##   traffic.risk    tx .fitted .resid    .hat .sigma  .cooksd .std.resid
##          <dbl> <dbl>   <dbl>  <dbl>   <dbl>  <dbl>    <dbl>      <dbl>
## 1         1.86     1    2.17 -0.312 0.00602  0.780 0.000490     -0.402
## 2         1        1    2.17 -1.17  0.00602  0.777 0.00687      -1.51
## 3         3.29     1    2.17  1.12  0.00602  0.777 0.00626       1.44
## 4         2        1    2.17 -0.170 0.00602  0.780 0.000144     -0.218
## 5         2.43     1    2.17  0.259 0.00602  0.780 0.000337      0.334
## 6         3.29     1    2.17  1.12  0.00602  0.777 0.00626       1.44
```

- `tidy` = model components like b
- `glance` is similar but for model fit measures; η^2 or R^2
- `augment` = adds to existing dataset

```
tidy(mod.1)
```

```
## # A tibble: 2 × 5
##   term        estimate std.error stat
##   <chr>          <dbl>     <dbl>
## 1 (Intercept)    2.65     0.0764
## 2 tx            -0.481    0.0974
```

```
glance(mod.1)
```

```
## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma stati
##       <dbl>         <dbl> <dbl>
## 1    0.0834        0.0800 0.779
## # ℹ 3 more variables: deviance <dbl>,
```

Pop quiz #2

- For a four group Oneway ANOVA how many different predicted values will we have? Residuals?
- Post your answer in Slack and see how you compare to your peers!

Statistical Inference

- In making predictions, we have to compare our prediction to some alternative prediction to see if we are doing well or not.
- What is our best guess (ie prediction) if we didn't collect any data?

$$\hat{Y} = \bar{Y}$$

- Regression can be thought of as: is $E(Y|X)$ better than $E(Y)$?

Statistical Inference

- To the extent that we can generate different predicted values of Y based on the values of the predictors, our model is doing well
- The closer our model is to the "actual" data generating model, our guesses ($\hat{Y}$) will be closer to our actual data (Y)

- We formally test how well we are doing with our guesses by partitioning variation

$$Y = \hat{Y} + e$$

$$Y = \hat{Y} + (Y - \hat{Y})$$

$$Y - \bar{Y} = (\hat{Y} - \bar{Y}) + (Y - \hat{Y})$$

$$(Y - \bar{Y})^2 = [(\hat{Y} - \bar{Y}) + (Y - \hat{Y})]^2$$

$$\sum (Y - \bar{Y})^2 = \sum (\hat{Y} - \bar{Y})^2 + \sum (Y - \hat{Y})^2$$

Partitioning the variation in Y

$$\sum (Y_i - \bar{Y})^2 = \sum (\hat{Y}_i - \bar{Y})^2 + \sum (Y_i - \hat{Y}_i)^2$$

- SS total = SS between + SS within
- SS total = SS regression + SS residual (or error)
- Completely the same as last semester because ANOVA IS REGRESSION

What can we do with this?

- Last semester you did omnibus F tests
- What hypothesis does the omnibus F test test, generally?

$$s_y^2 = s_{regression}^2 + s_{residual}^2$$

$$1 = \frac{s_{regression}^2}{s_y^2} + \frac{s_{residual}^2}{s_y^2}$$

Coefficient of Determination

$$\frac{s^2_{regression}}{s^2_y} = \frac{SS_{regression}}{SS_Y} = R^2$$

- Percent (of total) variance explained by your model...which currently are groups
- Another way of asking how much variance group status explains

R^2 and Eta squared

```
summary(mod.1)$r.squared
```

```
## [1] 0.08344007
```

```
library(lsr)
etaSquared(mod.1)
```

```
##         eta.sq eta.sq.part
## tx 0.08344007  0.08344007
```

R^2 for different coding schemes

```
glance(mod.1)
```

```
## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic    p.value    df logLik
##       <dbl>         <dbl> <dbl>     <dbl>      <dbl> <dbl>  <dbl>
## 1    0.0834        0.0800 0.779      24.4 0.00000138     1  -315.
## # ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int
```

```
glance(mod.1.eff)
```

```
## # A tibble: 1 × 12
##   r.squared adj.r.squared sigma statistic    p.value    df logLik
##       <dbl>         <dbl> <dbl>     <dbl>      <dbl> <dbl>  <dbl>
## 1    0.0834        0.0800 0.779      24.4 0.00000138     1  -315.
## # ℹ 3 more variables: deviance <dbl>, df.residual <int>, nobs <int
```

Note the R^2 p-value

```
##
## Call:
## lm(formula = traffic.risk ~ tx, data = example.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.65064 -0.59811 -0.02668  0.54475  2.54475
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  2.65064    0.07637  34.707 < 0.0000000000000002 ***
## tx          -0.48111    0.09740  -4.939              0.00000138 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7789 on 268 degrees of freedom
## Multiple R-squared:  0.08344,    Adjusted R-squared:  0.08002
## F-statistic:  24.4 on 1 and 268 DF,  p-value: 0.000001381
```

Summary


Always has been
t-test
ANOVA
correlations
MLM
SEM
Wait it's all regression

Summary

- We are using linear models to do the exact same tests as *t*-tests and ANOVAs
- It is the exact same because *t*-tests and ANOVAs are part of the general linear model
- Using linear models gives us the same information, and more!
- Provides a more systematic way at 1) building and testing your theoretical model and 2) comparing between alternative theoretical models
- You can get 1) estimates and 2) fit statistics from the model. Both are important.

Next time

Revisiting correlations, and turning those into regression