Multiple Linear Regression

DATA 606 - Statistics & Probability for Data Analytics

Jason Bryer, Ph.D. and Angela Lui, Ph.D.

April 17, 2024

One Minute Paper Results

What was the most important thing you learned during this class?

What important question remains unanswered for you?

Weight of Books

```
allbacks <- read.csv('../course_data/allbacks.csv')
head(allbacks)
```

```
##   X volume area weight cover
## 1 1    885  382    800    hb
## 2 2   1016  468    950    hb
## 3 3   1125  387   1050    hb
## 4 4    239  371    350    hb
## 5 5    701  371    750    hb
## 6 6    641  367    600    hb
```

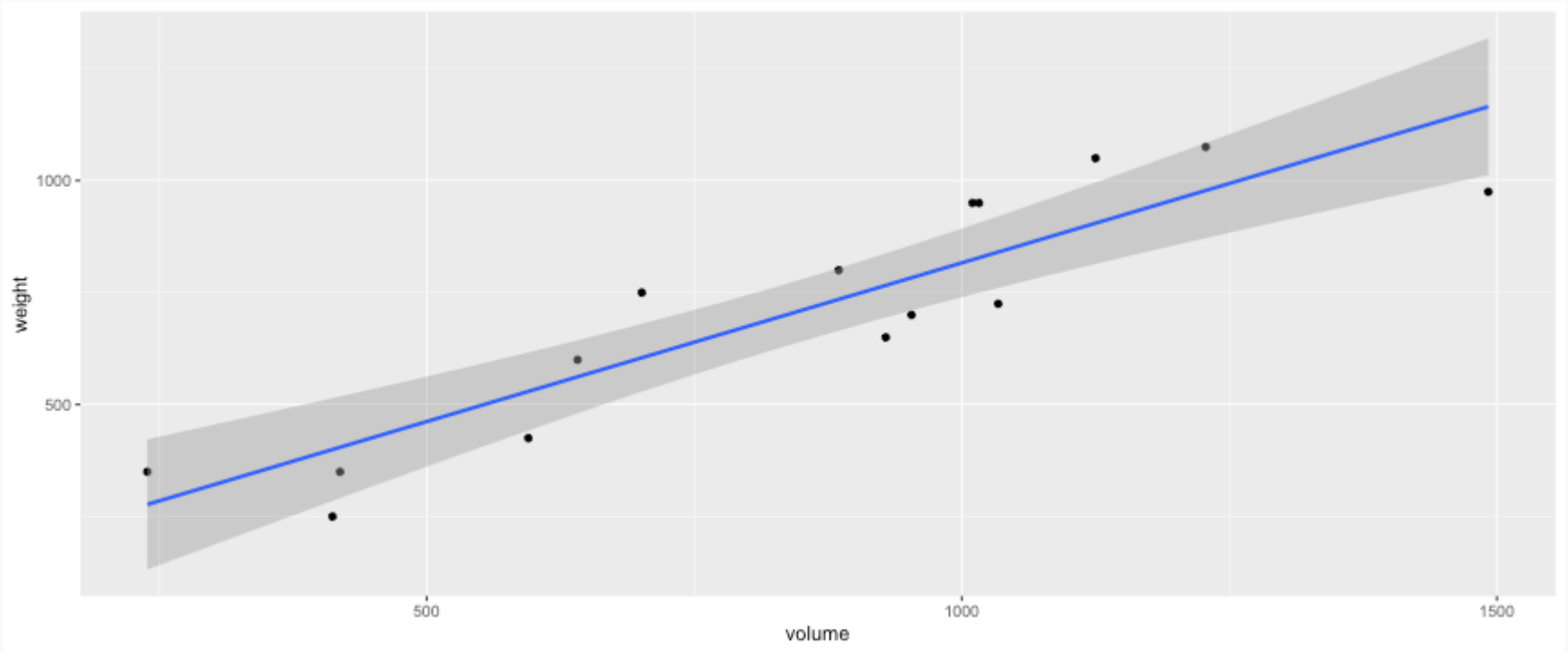
From: Maindonald, J.H. & Braun, W.J. (2007). *Data Analysis and Graphics Using R, 2nd ed.*

Weights of Books (cont)

```
lm.out <- lm(weight ~ volume, data=allbacks)
```

$$\hat{weight} = 108 + 0.71 volume$$

$$R^2 = 80\%$$



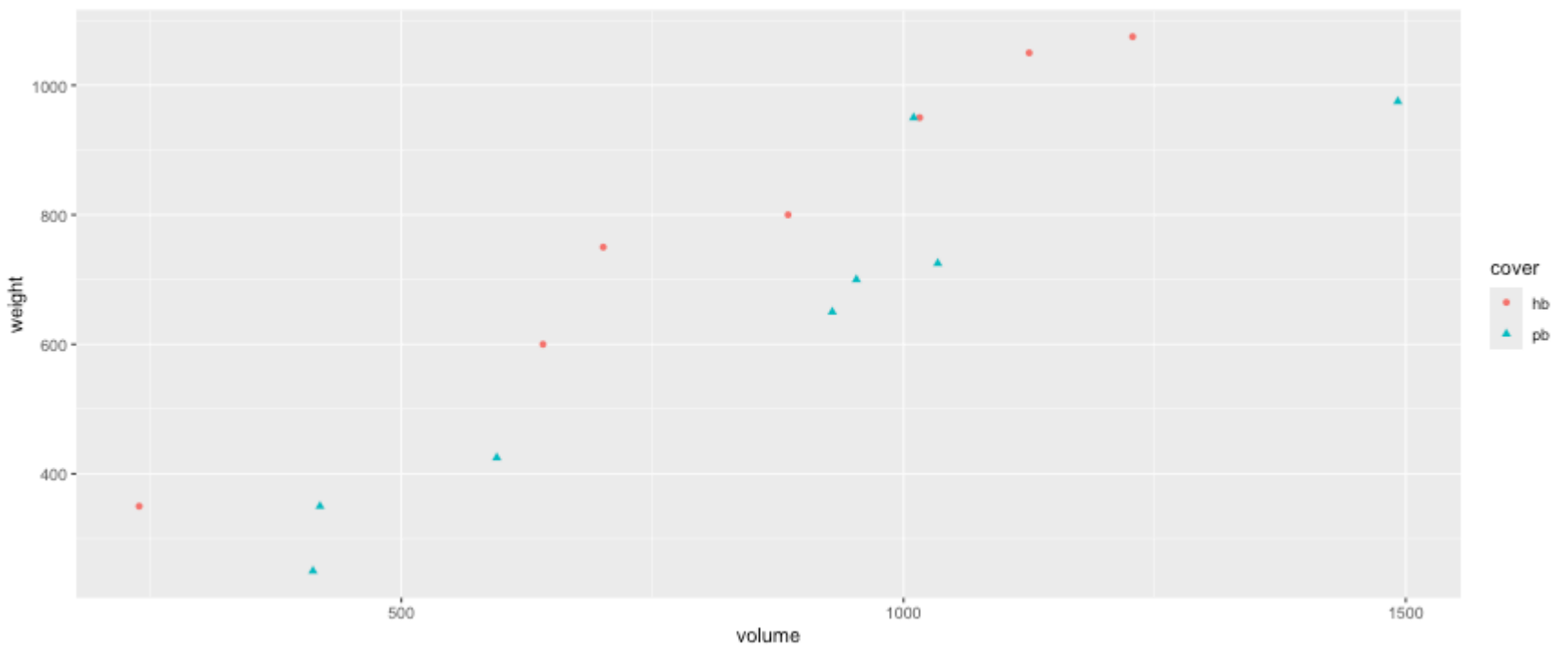
Modeling weights of books using volume

```
summary(lm.out)
```

```
##
## Call:
## lm(formula = weight ~ volume, data = allbacks)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -189.97 -109.86   38.08  109.73  145.57
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 107.67931   88.37758   1.218    0.245
## volume        0.70864    0.09746   7.271 6.26e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 123.9 on 13 degrees of freedom
## Multiple R-squared:  0.8026,    Adjusted R-squared:  0.7875
## F-statistic: 52.87 on 1 and 13 DF,  p-value: 6.262e-06
```

Weights of hardcover and paperback books

- Can you identify a trend in the relationship between volume and weight of hardcover and paperback books?

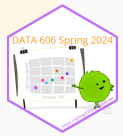


- Paperbacks generally weigh less than hardcover books after controlling for book's volume.

Modeling using volume and cover type

```
lm.out2 <- lm(weight ~ volume + cover, data=allbacks)
summary(lm.out2)
```

```
##
## Call:
## lm(formula = weight ~ volume + cover, data = allbacks)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -110.10  -32.32  -16.10   28.93  210.95
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  197.96284   59.19274   3.344 0.005841 **
## volume         0.71795    0.06153  11.669  6.6e-08 ***
## coverpb     -184.04727   40.49420  -4.545 0.000672 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.2 on 12 degrees of freedom
## Multiple R-squared:  0.9275,    Adjusted R-squared:  0.9154
## F-statistic: 76.73 on 2 and 12 DF,  p-value: 1.455e-07
```



Linear Model

$$\hat{weight} = 198 + 0.72volume - 184coverpb$$

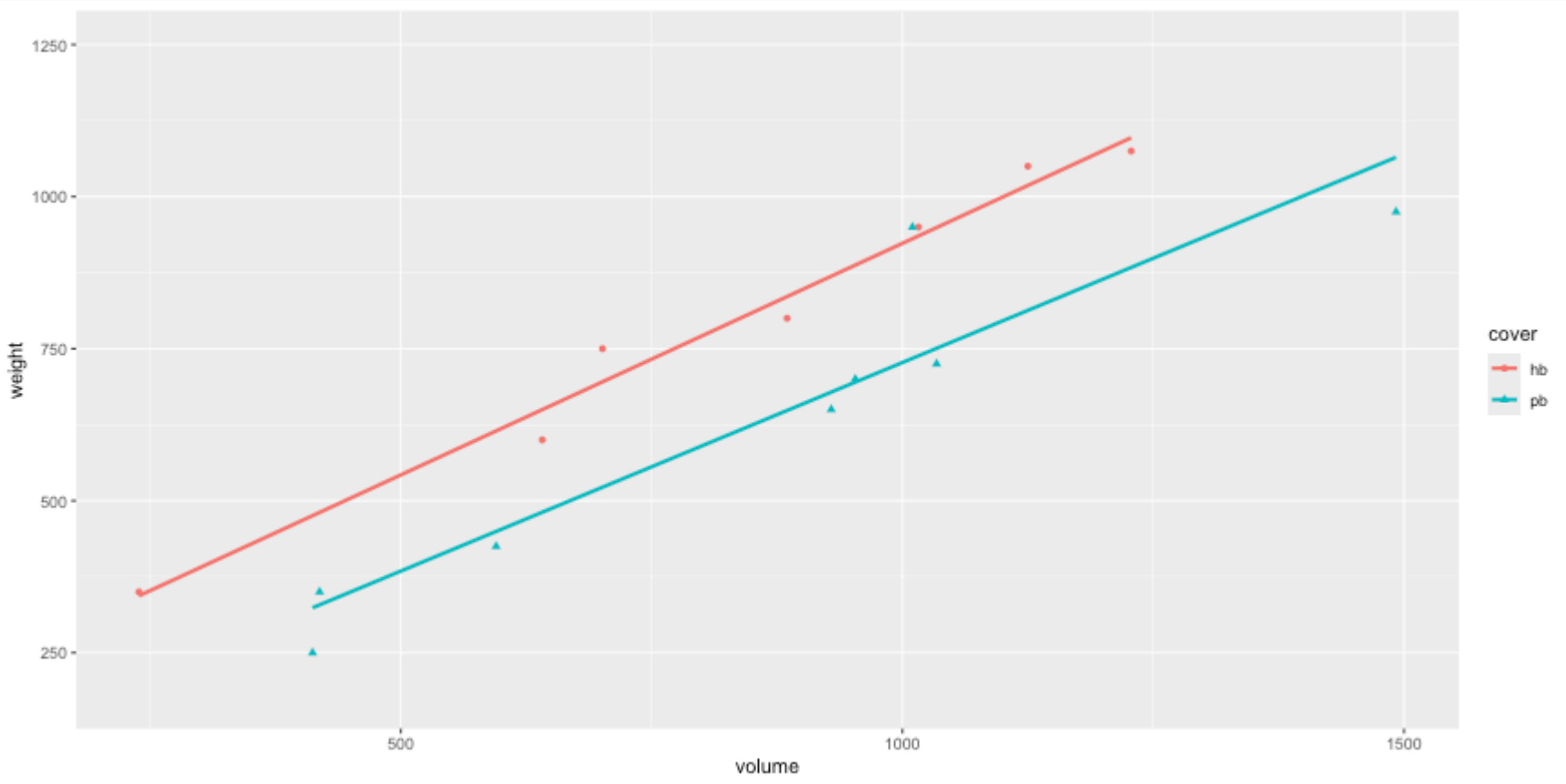
1. For **hardcover** books: plug in *0* for cover.

$$\hat{weight} = 197.96 + 0.72volume - 184.05 \times 0 = 197.96 + 0.72volume$$

1. For **paperback** books: put in 1 for cover.

$$\hat{weight} = 197.96 + 0.72volume - 184.05 \times 1$$

Visualizing the linear model



Interpretation of the regression coefficients

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	197.9628	59.1927	3.34	0.0058
volume	0.7180	0.0615	11.67	0.0000
coverpb	-184.0473	40.4942	-4.55	0.0007

- **Slope of volume**: All else held constant, books that are 1 more cubic centimeter in volume tend to weigh about 0.72 grams more.
- **Slope of cover**: All else held constant, the model predicts that paperback books weigh 184 grams lower than hardcover books.
- **Intercept**: Hardcover books with no volume are expected on average to weigh 198 grams.
  - Obviously, the intercept does not make sense in context. It only serves to adjust the height of the line.

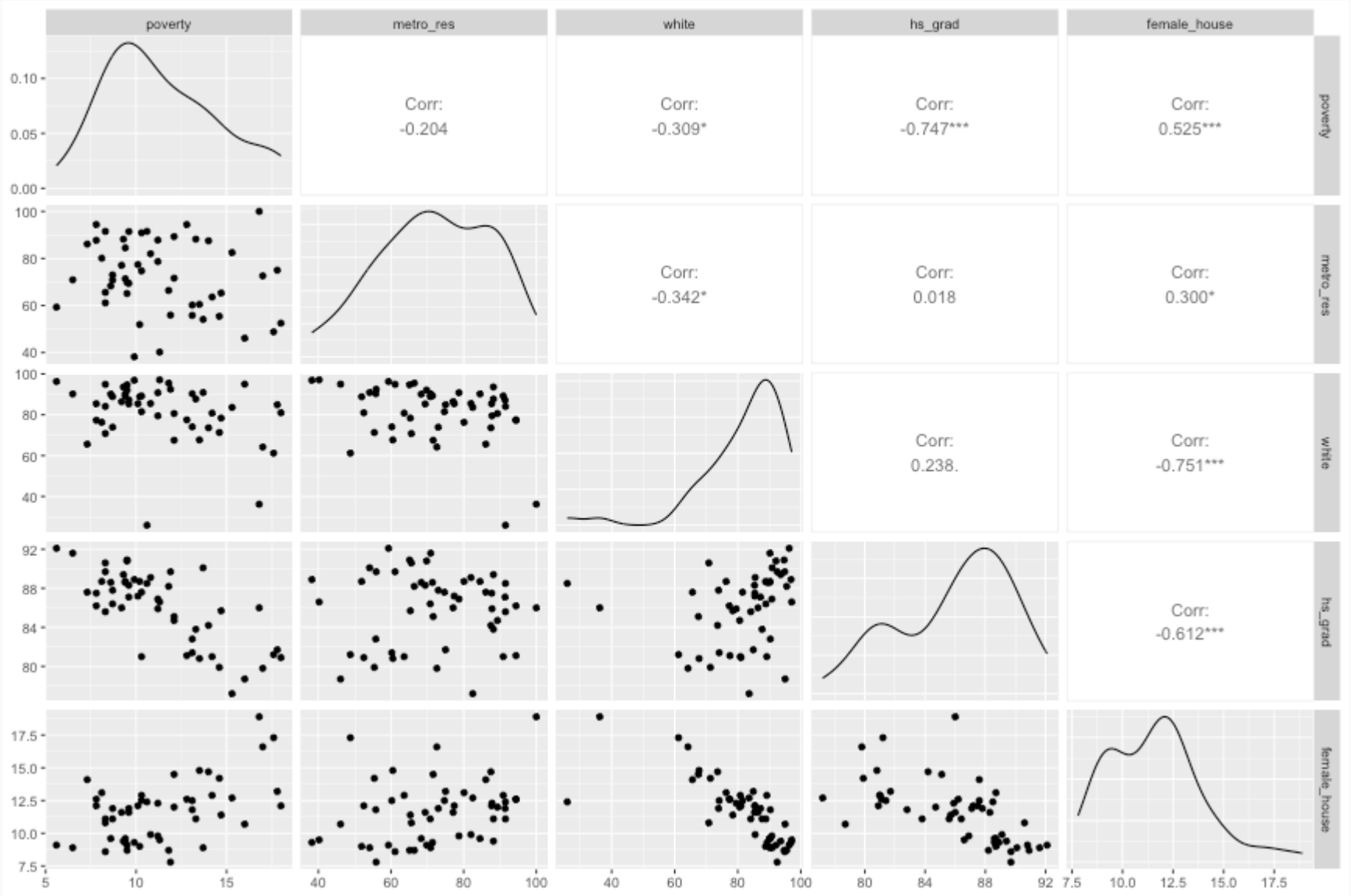
Modeling Poverty

```
poverty <- read.table("../course_data/poverty.txt", h = T, sep = "\t")
names(poverty) <- c("state", "metro_res", "white", "hs_grad", "poverty", "female_house")
poverty <- poverty[,c(1,5,2,3,4,6)]
head(poverty)
```

```
##        state poverty metro_res white hs_grad female_house
## 1    Alabama    14.6      55.4  71.3    79.9         14.2
## 2     Alaska     8.3      65.6  70.8    90.6         10.8
## 3    Arizona    13.3      88.2  87.7    83.8         11.1
## 4   Arkansas    18.0      52.5  81.0    80.9         12.1
## 5 California    12.8      94.4  77.5    81.1         12.6
## 6   Colorado     9.4      84.5  90.2    88.7          9.6
```

From: Gelman, H. (2007). *Data Analysis using Regression and Multilevel/Hierarchial Models.* Cambridge University Press.

Modeling Poverty

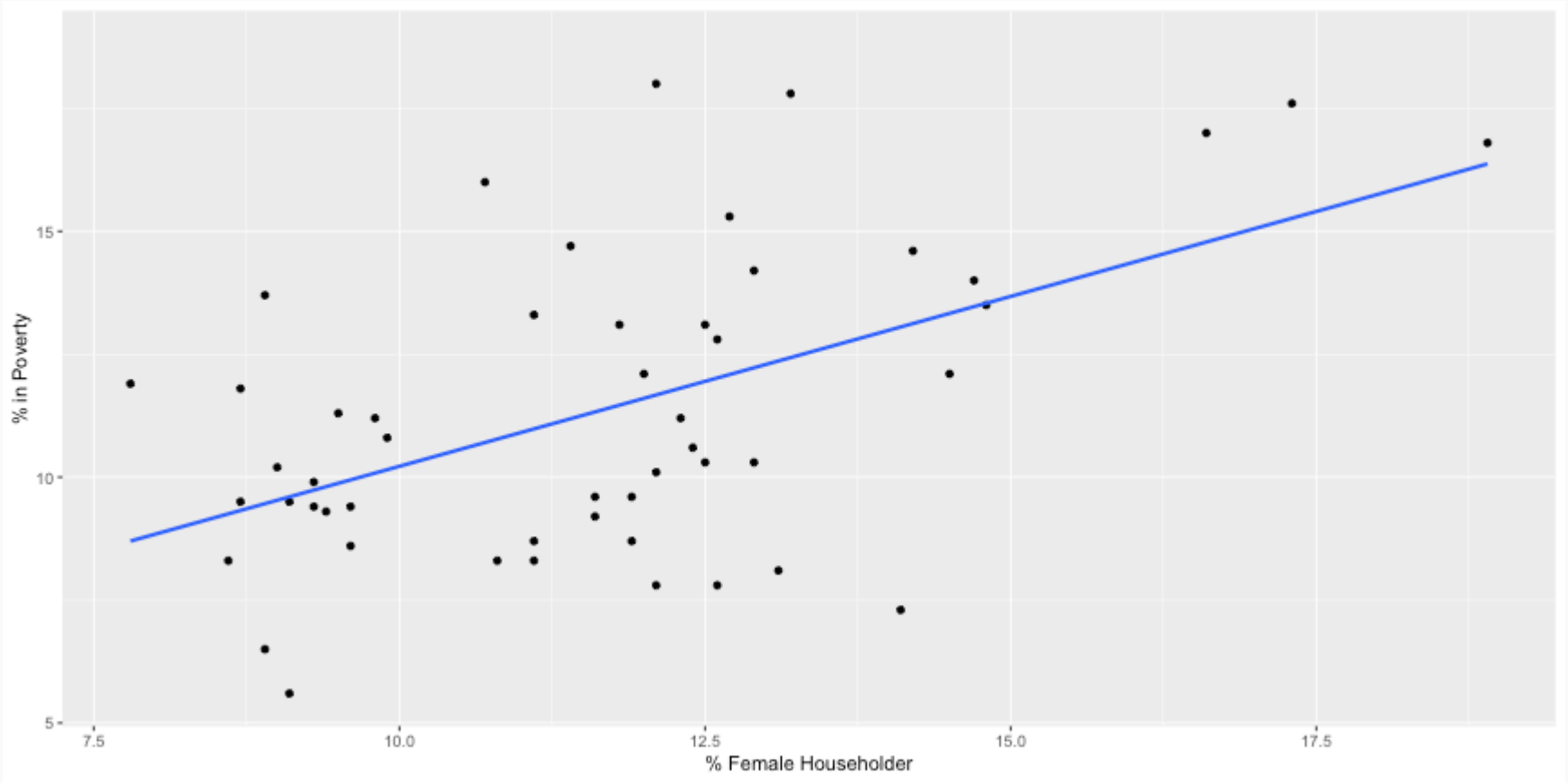


Predicting Poverty using Percent Female Householder

```
lm.poverty <- lm(poverty ~ female_house, data=poverty)
summary(lm.poverty)
```

```
##
## Call:
## lm(formula = poverty ~ female_house, data = poverty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.7537 -1.8252 -0.0375  1.5565  6.3285
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.3094     1.8970   1.745   0.0873 .
## female_house   0.6911     0.1599   4.322 7.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.664 on 49 degrees of freedom
## Multiple R-squared:  0.276,    Adjusted R-squared:  0.2613
## F-statistic: 18.68 on 1 and 49 DF,  p-value: 7.534e-05
```

% Poverty by % Female Household



Another look at R^2

R^2 can be calculated in three ways:

1. square the correlation coefficient of x and y (how we have been calculating it)
2. square the correlation coefficient of y and $\hat{y}$
3. based on definition:

$$R^2 = \frac{explained \quad variability \quad in \quad y}{total \quad variability \quad in \quad y}$$

Using ANOVA we can calculate the explained variability and total variability in y.

Sum of Squares

```
anova.poverty <- anova(lm.poverty)
print(xtable::xtable(anova.poverty, digits = 2), type='html')
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
female_house	1.00	132.57	132.57	18.68	0.00
Residuals	49.00	347.68	7.10		

Sum of squares of *y*: $SS_{Total} = \sum (y - \bar{y})^2 = 480.25 \rightarrow$ **total variability**

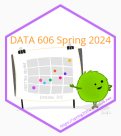
Sum of squares of residuals: $SS_{Error} = \sum e_i^2 = 347.68 \rightarrow$ **unexplained variability**

Sum of squares of *x*: $SS_{Model} = SS_{Total} - SS_{Error} = 132.57 \rightarrow$ **explained variability**

$$R^2 = \frac{explained \quad variability \quad in \quad y}{total \quad variability \quad in \quad y} = \frac{132.57}{480.25} = 0.28$$

Why bother?

- For single-predictor linear regression, having three ways to calculate the same value may seem like overkill.
- However, in multiple linear regression, we can't calculate R^2 as the square of the correlation between *x* and *y* because we have multiple *x*s.
- And next we'll learn another measure of explained variability, *adjusted* R^2, that requires the use of the third approach, ratio of explained and unexplained variability.



Predicting poverty using % female household & %

```
lm.poverty2 <- lm(poverty ~ female_house + white, data=poverty)
print(xtable::xtable(lm.poverty2), type='html')
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-2.5789	5.7849	-0.45	0.6577
female_house	0.8869	0.2419	3.67	0.0006
white	0.0442	0.0410	1.08	0.2868

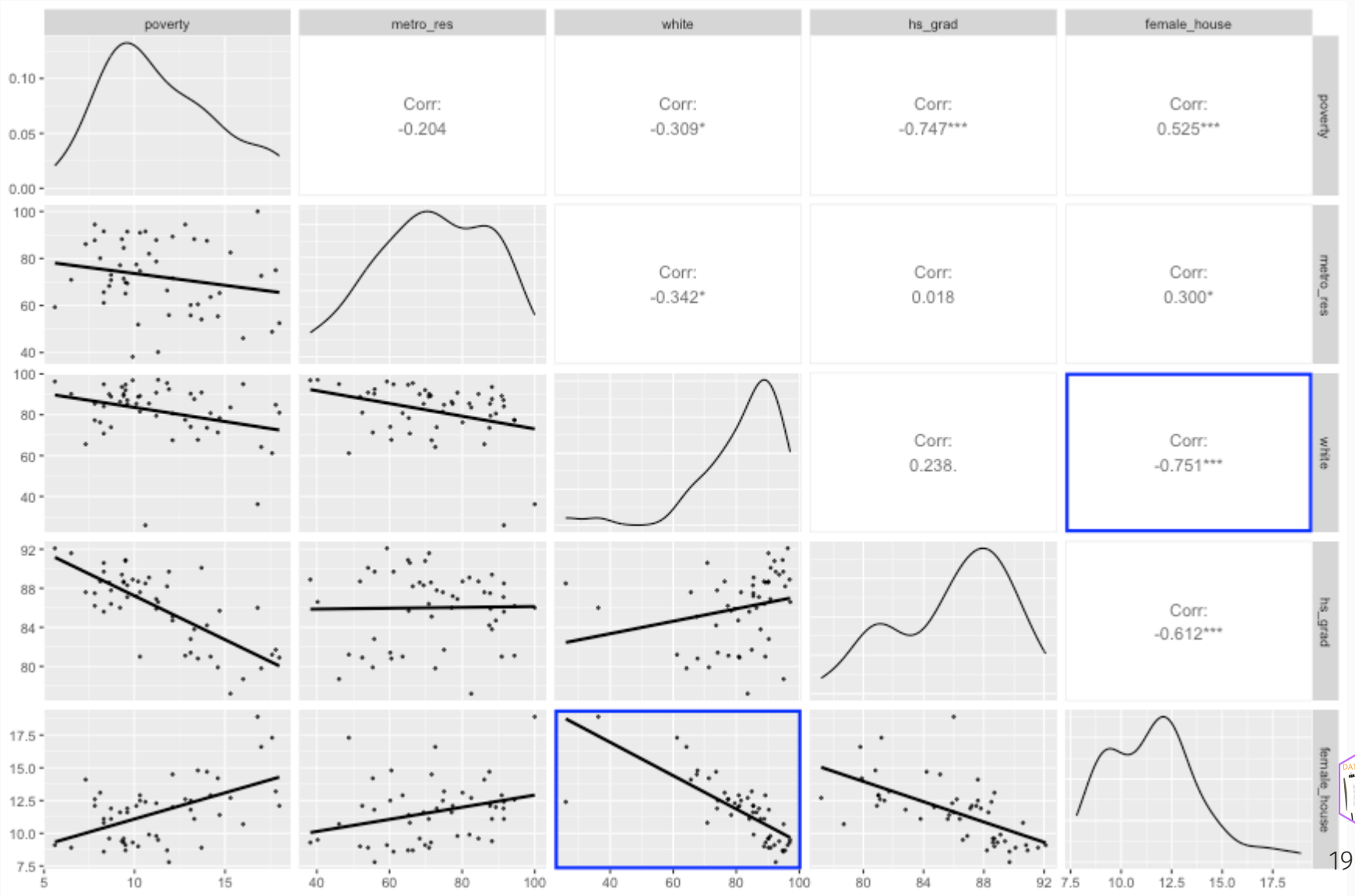
```
anova.poverty2 <- anova(lm.poverty2)
print(xtable::xtable(anova.poverty2, digits = 3), type='html')
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
female_house	1.000	132.568	132.568	18.745	0.000
white	1.000	8.207	8.207	1.160	0.287
Residuals	48.000	339.472	7.072		

$$R^2 = \frac{explained \quad variability \quad in \quad y}{total \quad variability \quad in \quad y} = \frac{132.57 + 8.21}{480.25} = 0.29$$

Unique information

Does adding the variable `white` to the model add valuable information that wasn't provided by `female_house`?



Collinearity between explanatory variables

poverty vs % female head of household

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.3094	1.8970	1.74	0.0873
female_house	0.6911	0.1599	4.32	0.0001

poverty vs % female head of household and % female household

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-2.5789	5.7849	-0.45	0.6577
female_house	0.8869	0.2419	3.67	0.0006
white	0.0442	0.0410	1.08	0.2868

Note the difference in the estimate for `female_house`.

Collinearity between explanatory variables

- Two predictor variables are said to be collinear when they are correlated, and this collinearity complicates model estimation.
  Remember: Predictors are also called explanatory or independent variables. Ideally, they would be independent of each other.

- We don't like adding predictors that are associated with each other to the model, because often times the addition of such variable brings nothing to the table. Instead, we prefer the simplest best model, i.e. *parsimonious* model.

- While it's impossible to avoid collinearity from arising in observational data, experiments are usually designed to prevent correlation among predictors

R^2 vs. adjusted R^2

Model	R^2	Adjusted R^2
Model 1 (Single-predictor)	0.28	0.26
Model 2 (Multiple)	0.29	0.26

- When any variable is added to the model R^2 increases.
- But if the added variable doesn't really provide any new information, or is completely unrelated, adjusted R^2 does not increase.

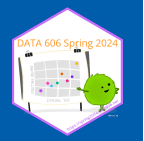
Adjusted R^2

$$R^2_{adj} = 1 - \left(\frac{SS_{error}}{SS_{total}} \times \frac{n-1}{n-p-1} \right)$$

where *n* is the number of cases and *p* is the number of predictors (explanatory variables) in the model.

- Because *p* is never negative, R^2_{adj} will always be smaller than R^2.
- R^2_{adj} applies a penalty for the number of predictors included in the model.
- Therefore, we choose models with higher R^2_{adj} over others.

Predictive Modeling



Example: Hours Studying Predicting Passing

```
study <- data.frame(
    Hours=c(0.50,0.75,1.00,1.25,1.50,1.75,1.75,2.00,2.25,2.50,2.75,3.00,
            3.25,3.50,4.00,4.25,4.50,4.75,5.00,5.50),
    Pass=c(0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,1,1,1,1,1)
)
study[sample(nrow(study), 5),]
```

```
##    Hours Pass
## 19  5.00    1
## 9   2.25    1
## 15  4.00    1
## 16  4.25    1
## 3   1.00    0
```

```
tab <- describeBy(study$Hours, group = study$Pass, mat = TRUE, skew = FALSE)
tab$group1 <- as.integer(as.character(tab$group1))
```

Prediction

Odds (or probability) of passing if studied **zero** hours?

$$log(\frac{p}{1-p}) = -4.078 + 1.505 \times 0$$

$$\frac{p}{1-p} = exp(-4.078) = 0.0169$$

$$p = \frac{0.0169}{1.169} = .016$$

Odds (or probability) of passing if studied **4** hours?

$$log(\frac{p}{1-p}) = -4.078 + 1.505 \times 4$$

$$\frac{p}{1-p} = exp(1.942) = 6.97$$

$$p = \frac{6.97}{7.97} = 0.875$$

Fitted Values

```
study[1,]
```

```
##   Hours Pass
## 1   0.5    0
```

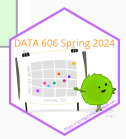
```
logistic <- function(x, b0, b1) {
    return(1 / (1 + exp(-1 * (b0 + b1 * x)) ))
}
logistic(.5, b0=-4.078, b1=1.505)
```

```
## [1] 0.03470667
```

Model Performance

The use of statistical models to predict outcomes, typically on new data, is called predictive modeling. Logistic regression is a common statistical procedure used for prediction. We will utilize a **confusion matrix** to evaluate accuracy of the predictions.

		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection, Power = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$	F_1 score = $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
		False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	Specificity (SPC), Selectivity, True negative rate (TNR) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$		



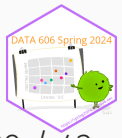
Predicting Heart Attacks

Source: https://www.kaggle.com/datasets/imnikhilanand/heart-attack-prediction?select=data.csv

```
heart <- read.csv('../course_data/heart_attack_predictions.csv')
heart <- heart |>
    mutate_if(is.character, as.numeric) |>
    select(!c(slope, ca, thal))
str(heart)
```

```
## 'data.frame':    294 obs. of  11 variables:
##  $ age     : int  28 29 29 30 31 32 32 32 33 34 ...
##  $ sex     : int  1 1 1 0 0 0 1 1 1 0 ...
##  $ cp      : int  2 2 2 1 2 2 2 2 3 2 ...
##  $ trestbps: num  130 120 140 170 100 105 110 125 120 130 ...
##  $ chol    : num  132 243 NA 237 219 198 225 254 298 161 ...
##  $ fbs     : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ restecg : num  2 0 0 1 1 0 0 0 0 0 ...
##  $ thalach : num  185 160 170 170 150 165 184 155 185 190 ...
##  $ exang   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ oldpeak : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ num     : int  0 0 0 0 0 0 0 0 0 0 ...
```

Note: `num` is the diagnosis of heart disease (angiographic disease status) (i.e. Value 0: < 50% diameter narrowing -- Value 1: > 50% diameter narrowing)



Missing Data

We will save this for another day...

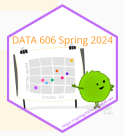
```
complete.cases(heart) |> table()
```

```
##
## FALSE  TRUE
##    33   261
```

```
mice_out <- mice::mice(heart, m = 1)
```

```
##
##  iter imp variable
##    1   1  trestbps  chol  fbs  restecg  thalach  exang
##    2   1  trestbps  chol  fbs  restecg  thalach  exang
##    3   1  trestbps  chol  fbs  restecg  thalach  exang
##    4   1  trestbps  chol  fbs  restecg  thalach  exang
##    5   1  trestbps  chol  fbs  restecg  thalach  exang
```

```
heart <- mice::complete(mice_out)
```



Data Setup

We will split the data into a training set (70% of observations) and validation set (30%).

```
train.rows <- sample(nrow(heart), nrow(heart) * .7)
heart_train <- heart[train.rows,]
heart_test <- heart[-train.rows,]
```

This is the proportions of survivors and defines what our "guessing" rate is. That is, if we guessed no one had a heart attack, we would be correct 62% of the time.

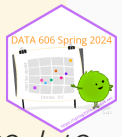
```
(heart_attack <- table(heart_train$num) %>% prop.table)
```

```
##
##         0         1
## 0.6243902 0.3756098
```

Model Training

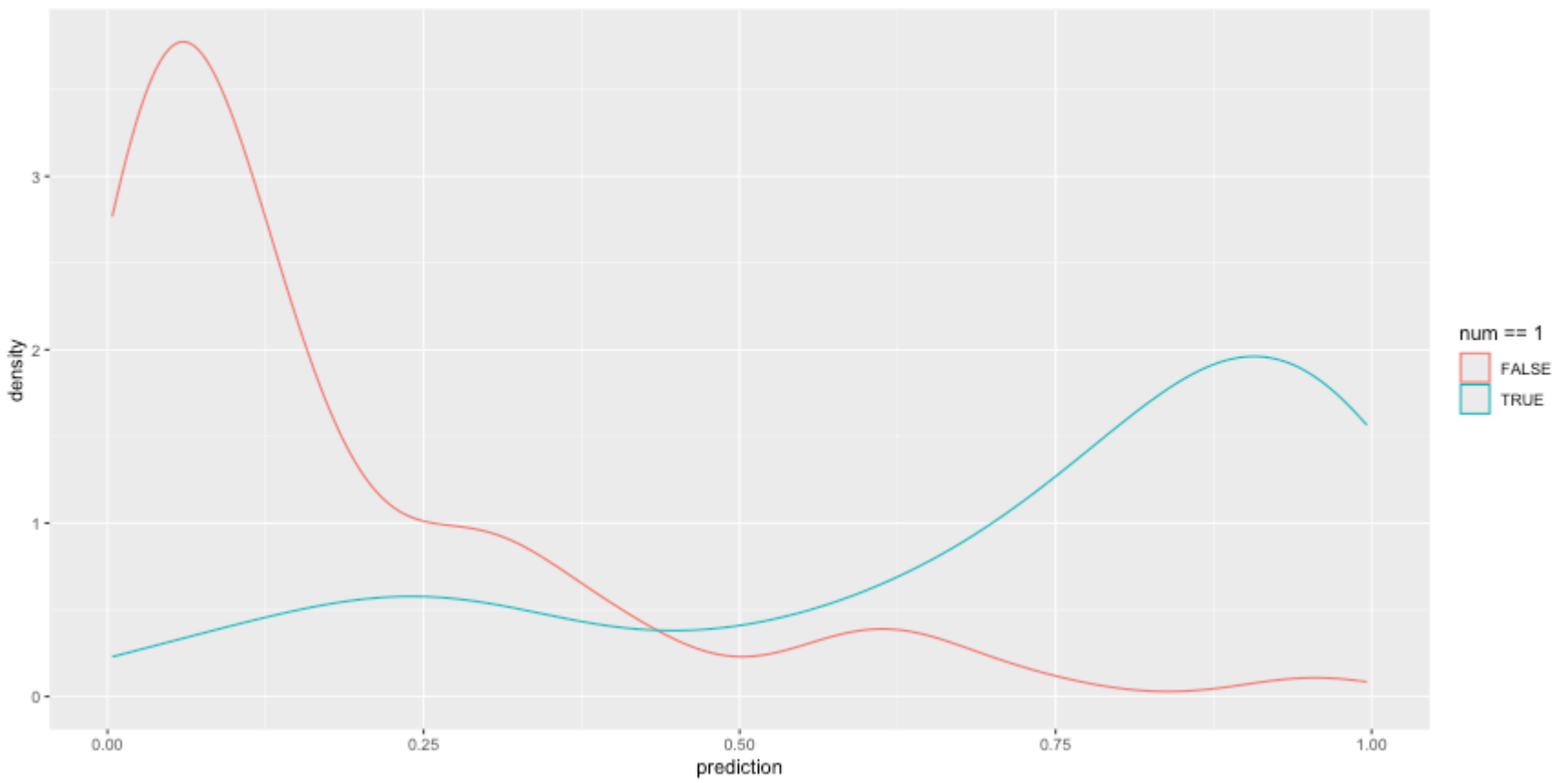
```
lr.out <- glm(num ~ ., data=heart_train, family=binomial(link = 'logit'))
summary(lr.out)
```

```
##
## Call:
## glm(formula = num ~ ., family = binomial(link = "logit"), data = heart_train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.177722   3.463954  -1.206 0.227796
## age         -0.033228   0.033254  -0.999 0.317689
## sex          1.314556   0.563216   2.334 0.019595 *
## cp           0.934521   0.266109   3.512 0.000445 ***
## trestbps     0.008442   0.014618   0.578 0.563591
## chol         0.003300   0.003131   1.054 0.291781
## fbs          2.016907   0.789637   2.554 0.010643 *
## restecg     -0.167309   0.524195  -0.319 0.749594
## thalach     -0.013756   0.011142  -1.235 0.216970
## exang        1.065755   0.508887   2.094 0.036234 *
## oldpeak      1.027296   0.273848   3.751 0.000176 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 271.37  on 204  degrees of freedom
## Residual deviance: 147.41  on 194  degrees of freedom
## AIC: 169.41
##
```



Predicted Values

```
heart_train$prediction <- predict(lr.out, type = 'response', newdata = heart_train)
ggplot(heart_train, aes(x = prediction, color = num == 1)) + geom_density()
```



Results

```
heart_train$prediction_class <- heart_train$prediction > 0.5
tab <- table(heart_train$prediction_class,
             heart_train$num) %>% prop.table() %>% print()
```

```
##
##                    0          1
##    FALSE 0.56585366 0.09268293
##    TRUE  0.05853659 0.28292683
```

For the training set, the overall accuracy is 84.88%. Recall that 62.44% people did not have a heart attach. Therefore, the simplest model would be to predict that no one had a heart attack, which would mean we would be correct 62.44% of the time. Therefore, our prediction model is 22.44% better than guessing.

Checking with the validation dataset

```
(survived_test <- table(heart_test$num) %>% prop.table())
```

```
##
##         0         1
## 0.6741573 0.3258427
```

```
heart_test$prediction <- predict(lr.out, newdata = heart_test, type = 'response')
heart_test$prediciton_class <- heart_test$prediction > 0.5
tab_test <- table(heart_test$prediciton_class, heart_test$num) %>%
    prop.table() %>% print()
```

```
##
##                   0          1
##   FALSE 0.62921348 0.11235955
##   TRUE  0.04494382 0.21348315
```

The overall accuracy is 84.27%, or 16.9% better than guessing.

Receiver Operating Characteristic (ROC) Curve

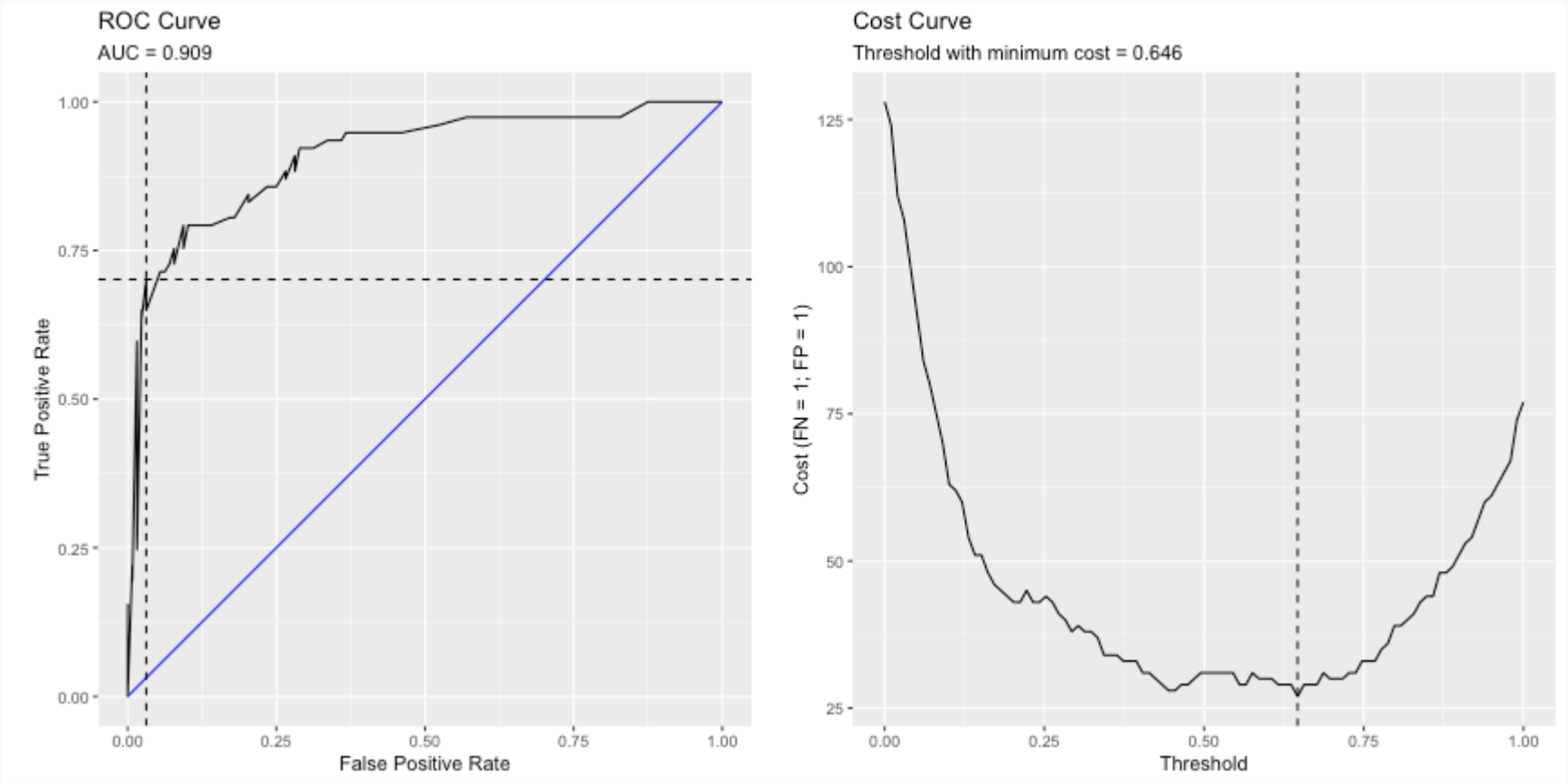
The ROC curve is created by plotting the true positive rate (TPR; AKA sensitivity) against the false positive rate (FPR; AKA probability of false alarm) at various threshold settings.

```
roc <- calculate_roc(heart_train$prediction, heart_train$num == 1)
summary(roc)
```

```
## AUC = 0.909
## Cost of false-positive = 1
## Cost of false-negative = 1
## Threshold with minimum cost = 0.646
```

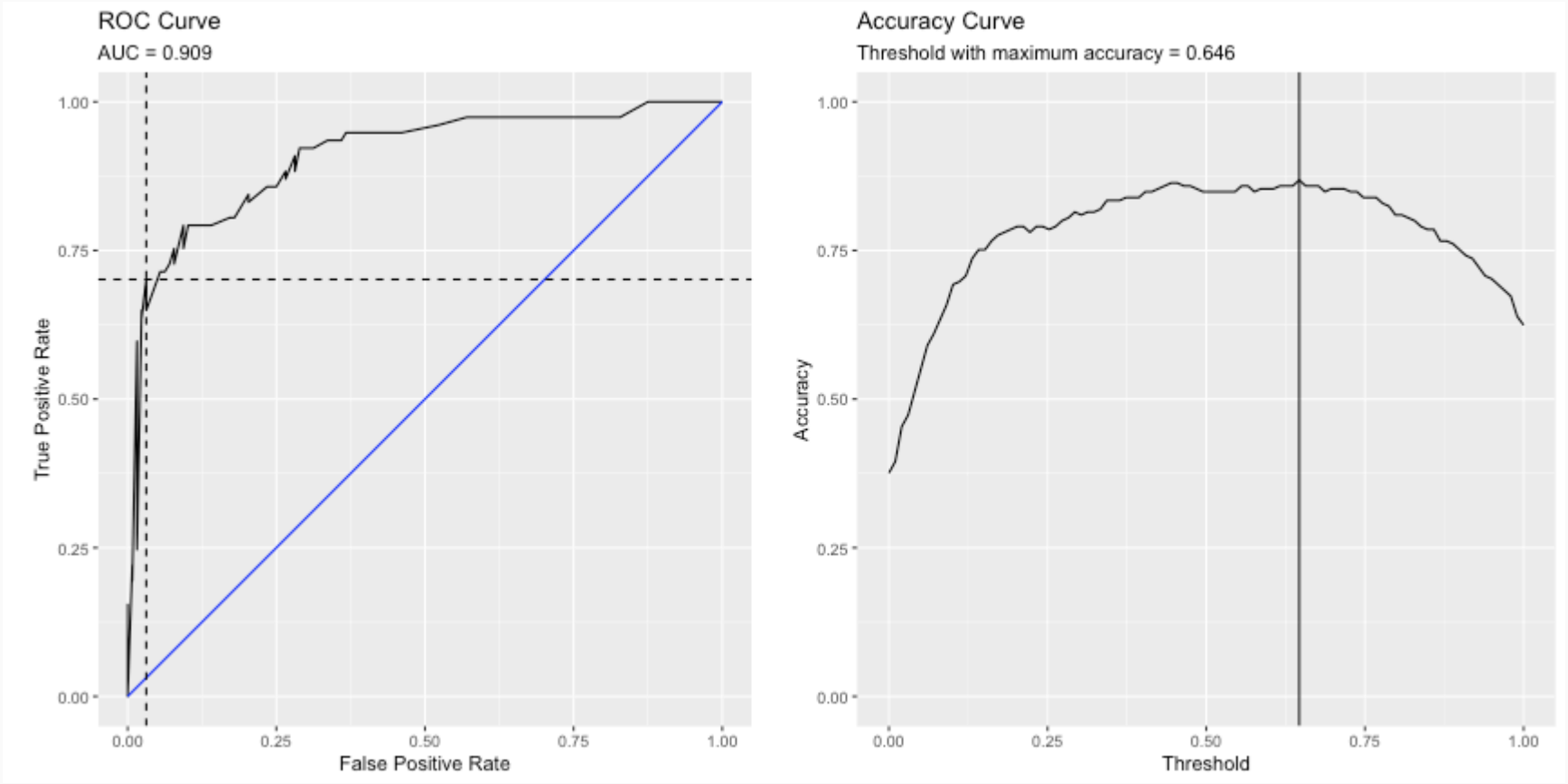
ROC Curve

```
plot(roc)
```



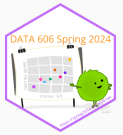
ROC Curve

```
plot(roc, curve = 'accuracy')
```



Caution on Interpreting Accuracy

- Loh, Sooo, and Zing (2016) predicted sexual orientation based on Facebook Status.
- They reported model accuracies of approximately 90% using SVM, logistic regression and/or random forest methods.
- Gallup (2018) poll estimates that 4.5% of the Americal population identifies as LGBT.
- *My proposed model:* I predict all Americans are heterosexual.
- The accuracy of my model is 95.5%, or *5.5% better than Facebook's model!*
- Predicting "rare" events (i.e. when the proportion of one of the two outcomes large) is difficult and requires independent (predictor) variables that strongly associated with the dependent (outcome) variable.



Fitted Values Revisited

What happens when the ratio of true-to-false increases (i.e. want to predict "rare" events)?

Let's simulate a dataset where the ratio of true-to-false is 10-to-1. We can also define the distribution of the dependent variable. Here, there is moderate separation in the distributions.

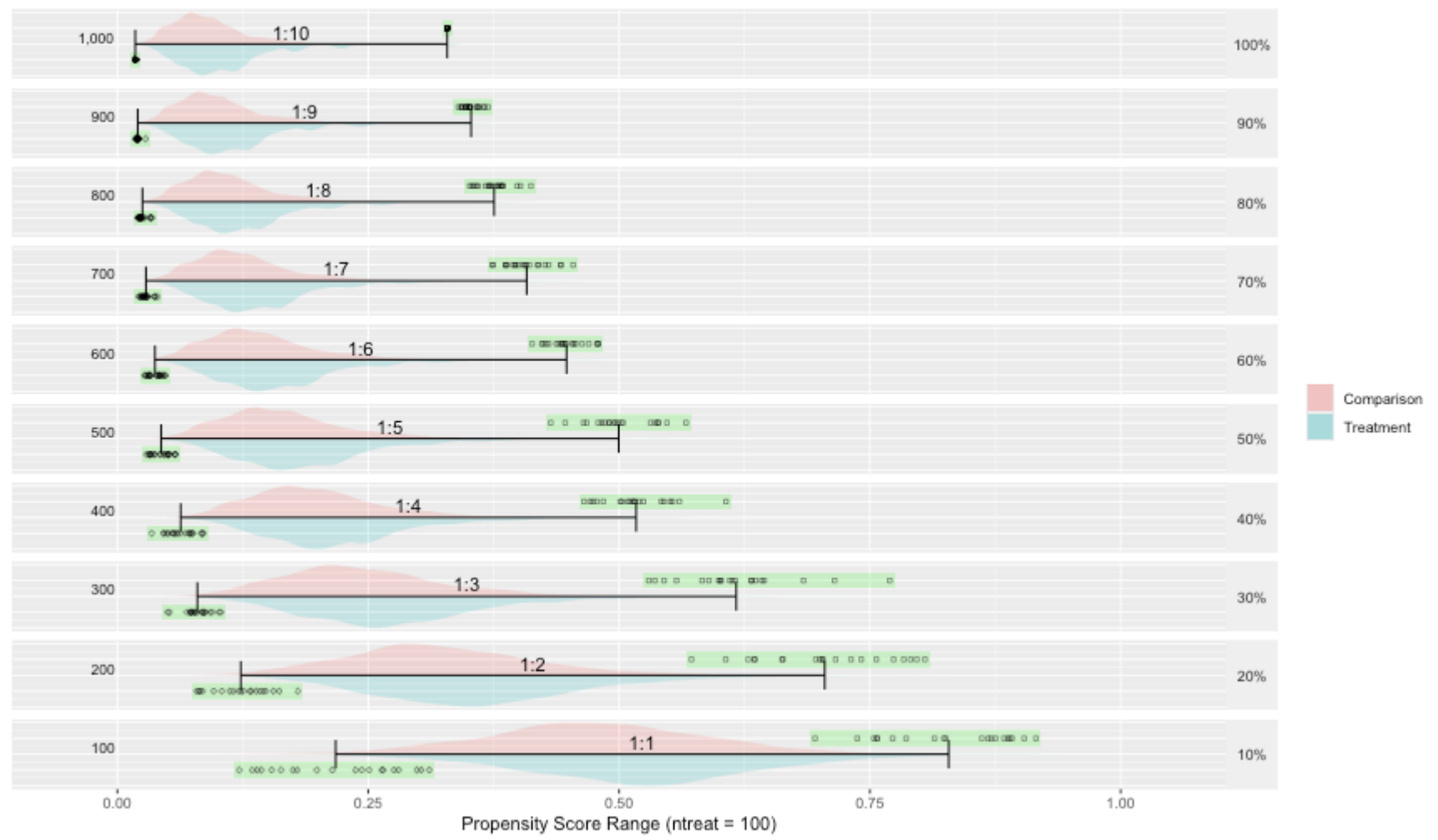
```
test.df2 <- getSimulatedData(
    treat.mean=.6, control.mean=.4)
```

The `multilevelPSA::psrange` function will sample with varying ratios from 1:10 to 1:1. It takes multiple samples and averages the ranges and distributions of the fitted values from logistic regression.

```
psranges2 <- psrange(test.df2, test.df2$treat, treat ~ .,
                     samples=seq(100,1000,by=100), nboot=20)
```

Fitted Values Revisited (cont.)

```
plot(psranges2)
```



Additional Resources

- The Path to Log Likelihood
- Visual Introduction to Maximum Likelihood Estimation
- VisualStats R Package
- Logistic Regression Details Pt 2: Maximum Likelihood
- StatQuest: Maximum Likelihood, clearly explained
- Probability concepts explained: Maximum likelihood estimation

One Minute Paper

Complete the one minute paper:

https://forms.gle/Jcw55CYvc6Ym8A5F7

1. What was the most important thing you learned during this class?
2. What important question remains unanswered for you?