Birth Weights

Oscar Briones Ramirez

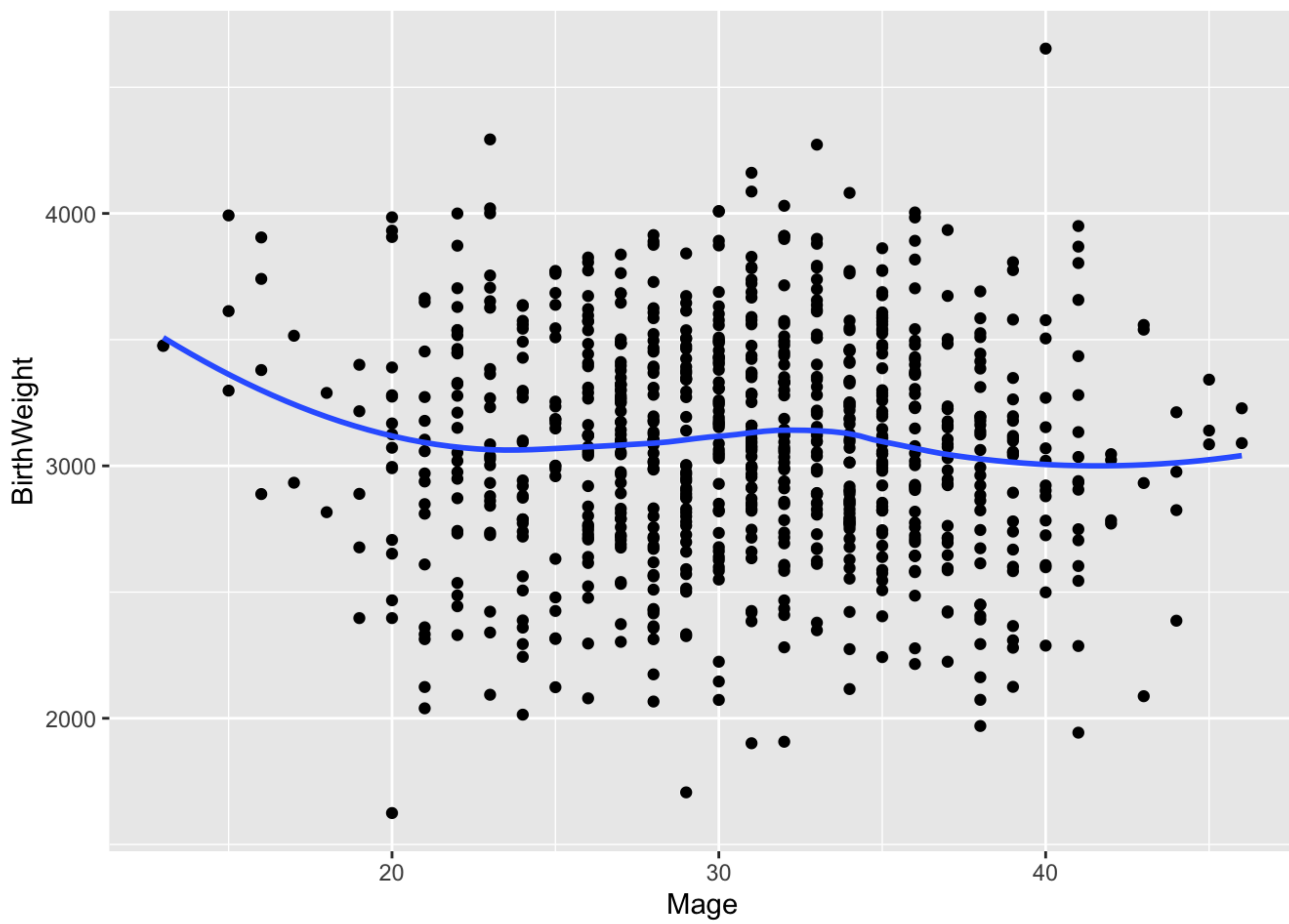
2023-01-30

EDA

1. Scatterplot of BirthWeight by Mage

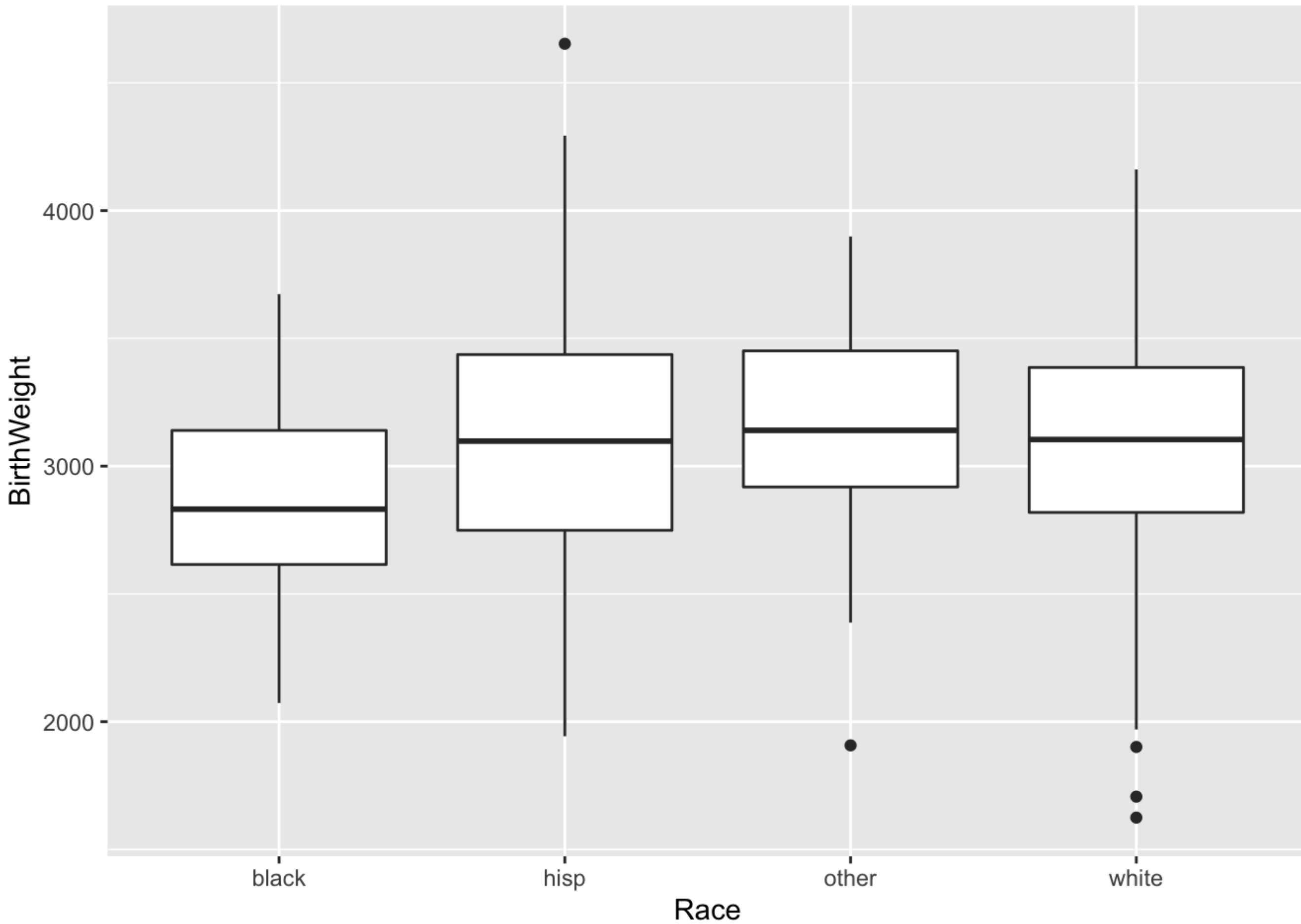
```
#1. Scatterplot BirthWeight by Mage
ggplot(data=birth_weights, mapping=aes(x=Mage, y=BirthWeight)) + geom_point()+geom_smoot
h(se=FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



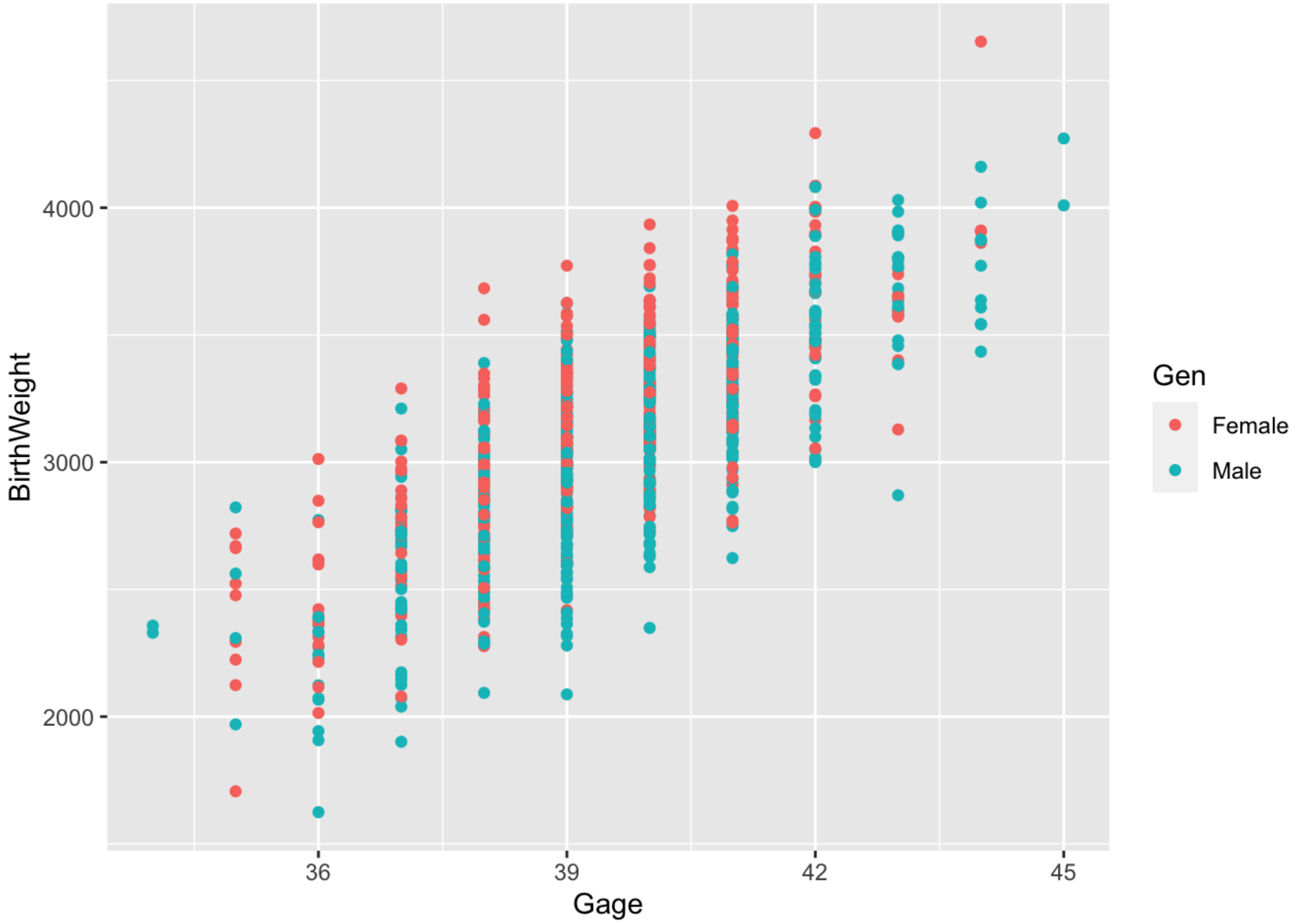
2. Side-by-side boxplots of BirthWeight for each category in Race

```
#2. boxplot
ggplot(data=birth_weights, mapping=aes(x=Race, y=BirthWeight)) + geom_boxplot()
```



3. A scatterplot of BirthWeight by Gage where the dots are colored according to Gen

```
#3. Scatterplot BirthWeight by Gage where the dots are colored according to Gen
ggplot(data=birth_weights, mapping=aes(x=Gage, y=BirthWeight, color = Gen)) + geom_point
()
```



4. The correlation between BirthWeight and Mage.

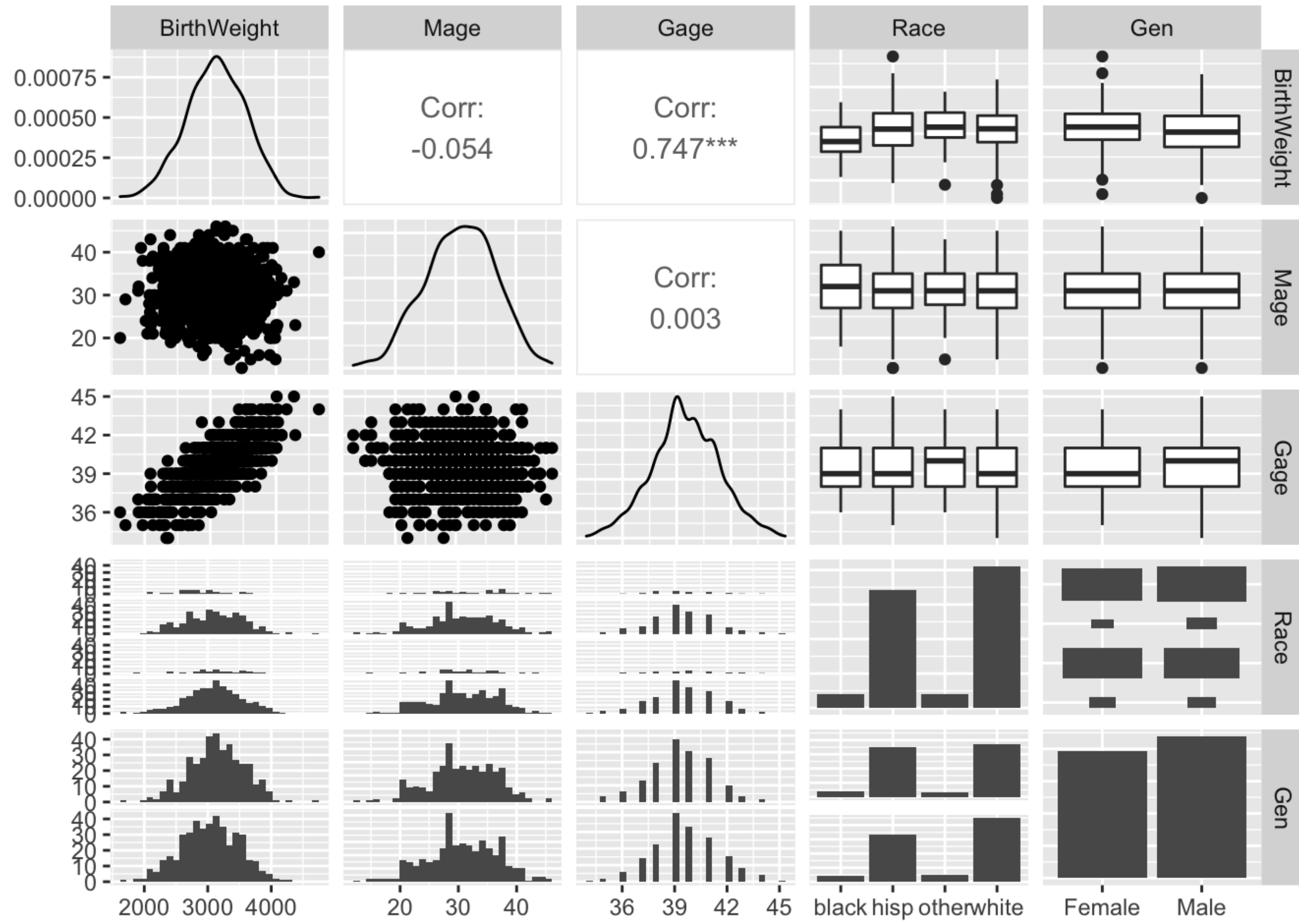
```
#4. The correlation between BirthWeight and Mage.
cor(birth_weights$BirthWeight, birth_weights$Mage)
```

```
## [1] -0.0537451
```

5. A pairs plot of all the variables in the BirthWeight dataset.

```
#5. Pairs plots

ggpairs(birth_weights)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Fitting a Linear Model

1. Without the use of lm() calculate β and s2. Verify your answer using lm().

```
Bhat <- solve((t(X)%*%X))%*%t(X)%*%y
coef(birth.lm)
```

```
##  (Intercept)         Mage         Gage     Racehisp    Raceother    Racewhite
## -4120.542409    -3.793751   182.742497   198.747954   241.582827   204.888197
##      GenMale
##  -169.348562
```

```
Bhat
```

```
##                      [,1]
## (Intercept) -4120.542409
## Mage           -3.793751
## Gage          182.742497
## Racehisp      198.747954
## Raceother     241.582827
## Racewhite     204.888197
## GenMale      -169.348562
```

```
S2 <- (t(y-(X%*%Bhat))%*%(y-(X%*%Bhat)))/(832-4-1)
sigma(birth.lm)
```

```
## [1] 281.5619
```

```
sqrt(S2)
```

```
##          [,1]
## [1,] 281.2212
```

2. Without the use of lm() calculate the fitted values Xβ . Verify your calculations by pulling off the fitted values from an lm() object.

```
fitvals <- X%*%Bhat
head(fitted(birth.lm))
```

```
##        1        2        3        4        5        6
## 2954.698 3074.728 2716.497 3502.926 2922.336 3245.756
```

```
head(fitvals)
```

```
##       [,1]
## 1 2954.698
## 2 3074.728
## 3 2716.497
## 4 3502.926
## 5 2922.336
## 6 3245.756
```

3. Without the use of lm() calculate the residuals y−Xβ Verify your calculations by pulling off the residuals from an lm() object.

```
resids <- y-X%*%Bhat
head(resid(birth.lm))
```

```
##          1          2          3          4          5          6
##   72.27167  158.38188  -72.10682  497.52418  359.15436 -496.87555
```

```
head(resids)
```

```
##          [,1]
## 1   72.27167
## 2  158.38188
## 3  -72.10682
## 4  497.52418
## 5  359.15436
## 6 -496.87555
```

4. Identify your model R2 from the summary() output.

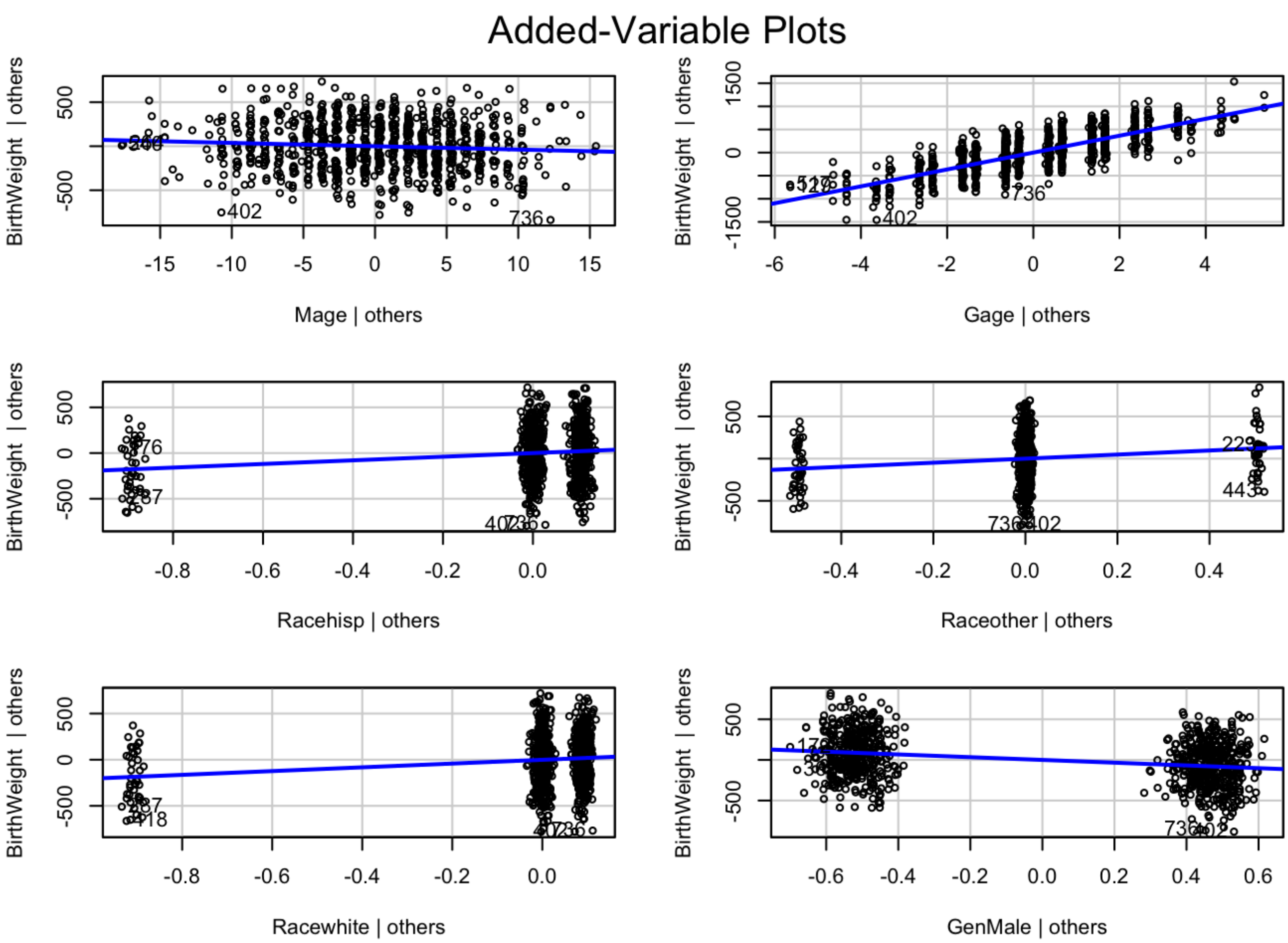
```
summary(birth.lm)$r.squared
```

```
## [1] 0.6064689
```

Checking Assumptions

1. Construct added variable plots and assess if the linearity assumption is OK for this data.

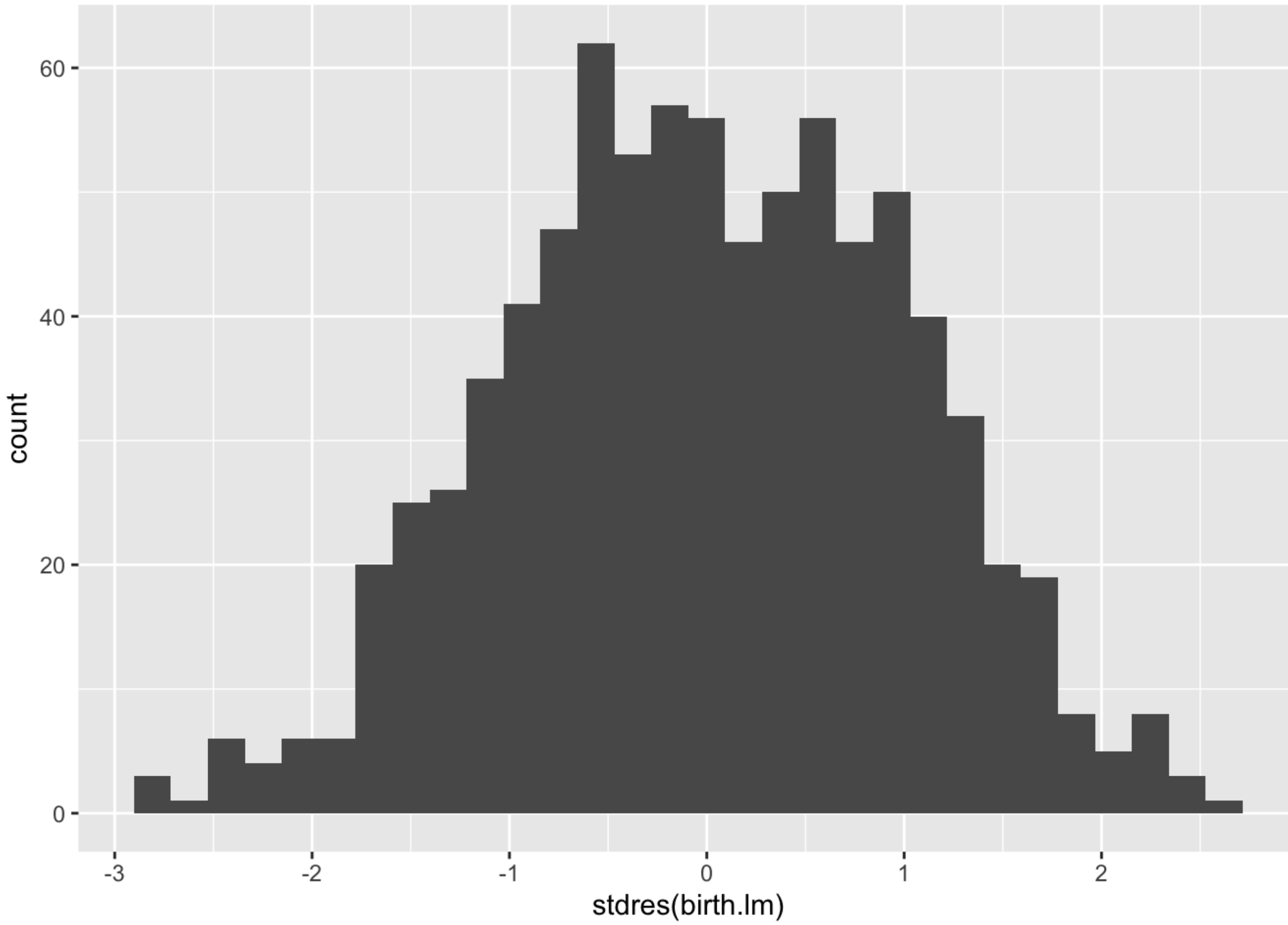
```
avPlots(birth.lm, ask=FALSE)
```



2. Construct a histogram of the standardized residuals and run a KS-test to see if the normality assumption is OK for this data.

```
ggplot() + geom_histogram(mapping=aes(x=stdres(birth.lm)))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

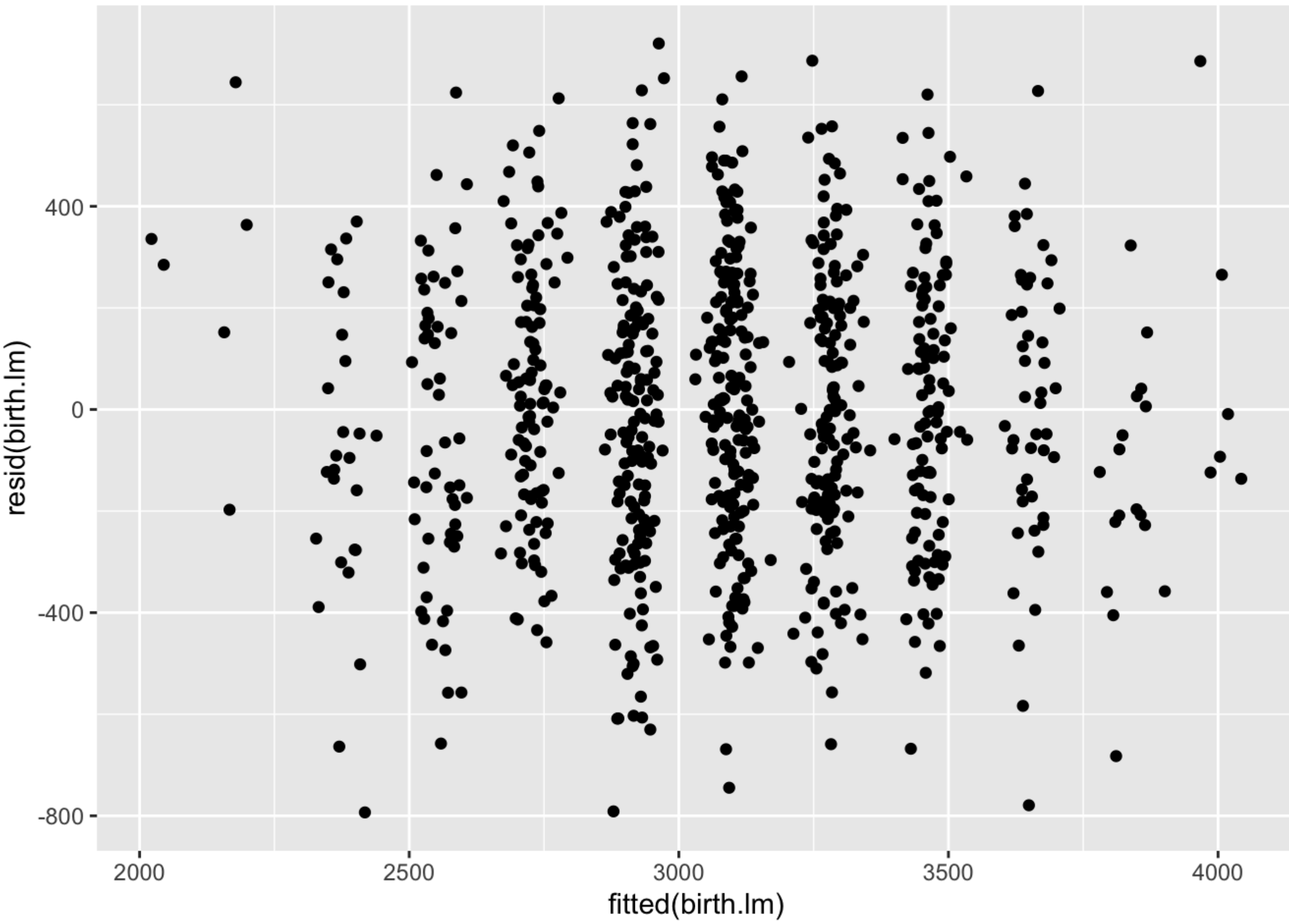



```
ks.test(stdres(birth.lm), "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  stdres(birth.lm)
## D = 0.028947, p-value = 0.4884
## alternative hypothesis: two-sided
```

3. Draw a scatterplot of the fitted values vs. standardized residuals and run a BP-test to see if the equal variance assumption is OK for this data.

```
ggplot(mapping=aes(x=fitted(birth.lm), y=resid(birth.lm))) + geom_point()
```




```
bptest(birth.lm)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  birth.lm
## BP = 6.8177, df = 6, p-value = 0.338
```

Predictions

1. Without using predict.lm(), calculate your point prediction of the birth weight for a baby with Mage=26,Gage=37, Race="hisp" and Gen="Female" using the formula ŷ new=xnewβ where β is the maximum likelihood estimate that you calculated above. Confirm that this is what predict.lm() is doing to get the point prediction.

```
newx <- data.frame(Intercept = 1, Mage=26, Gage=37, Racehisp=1, Raceother=0, Racewhite=
0, GenMale=0)
ynew <- newx*Bhat
rowSums(ynew)
```

```
## [1] 2741.04
```

```
new.x = data.frame(Mage=26, Gage=37, Race="hisp", Gen="Female")
predict.lm(birth.lm, newdata=new.x, interval="prediction", level=0.99)
```

```
##       fit      lwr      upr
## 1 2741.04 2011.669 3470.412
```

2. Using predict.lm(), get a prediction of the birth weight for a baby with Mage=26, Gage=37, Race="hisp" and Gen="Female" and an associated 99% prediction interval.

```
new.x = data.frame(Mage=26, Gage=37, Race="hisp", Gen="Female")
predict.lm(birth.lm, newdata=new.x, interval="prediction", level=0.99)
```

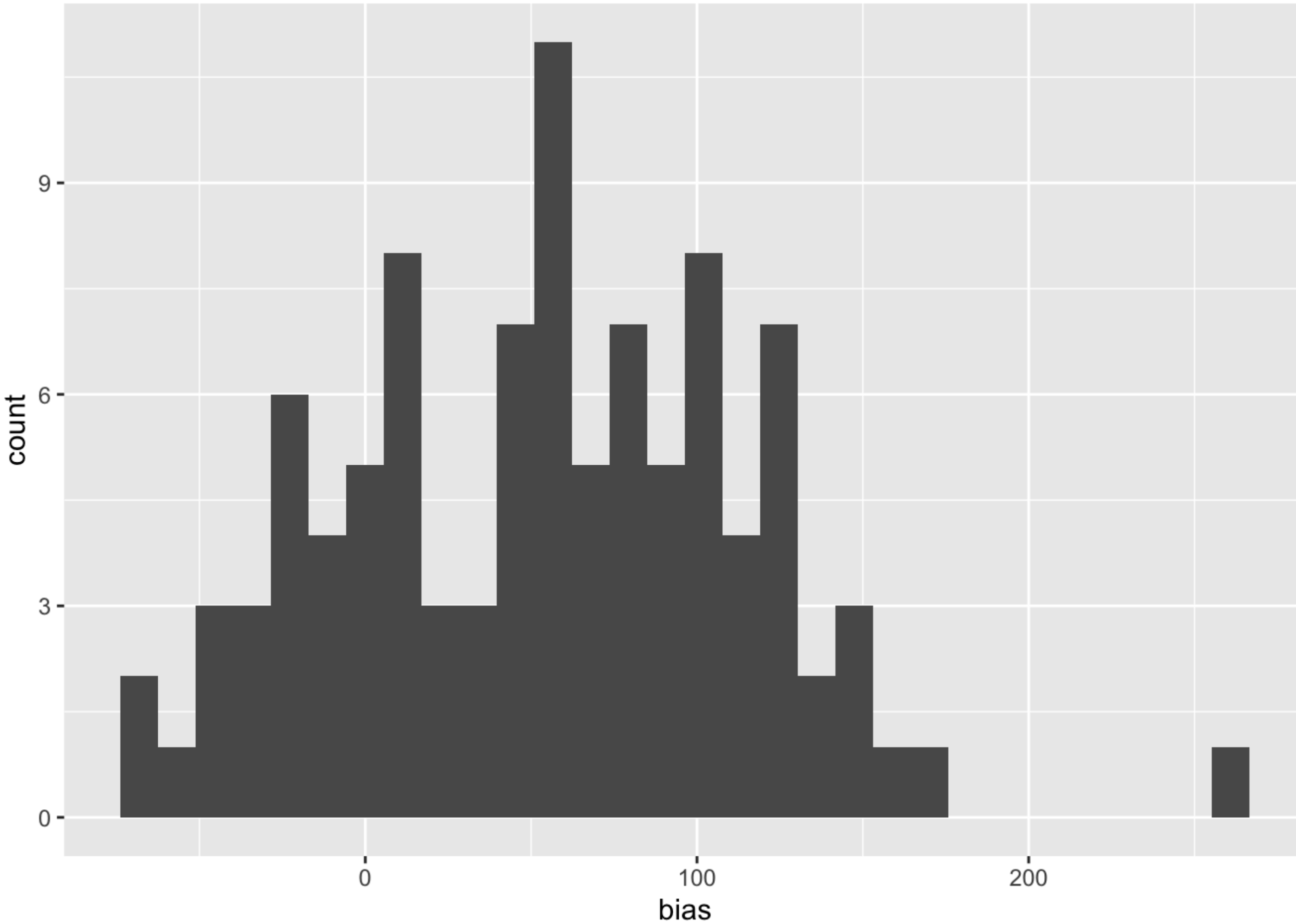
```
##       fit      lwr      upr
## 1 2741.04 2011.669 3470.412
```

Cross Validation

1. Adjust the above code to run 100 Monte Carlo cross validations and plot histograms (or density plots) of the bias, RPMSE, coverage and width.

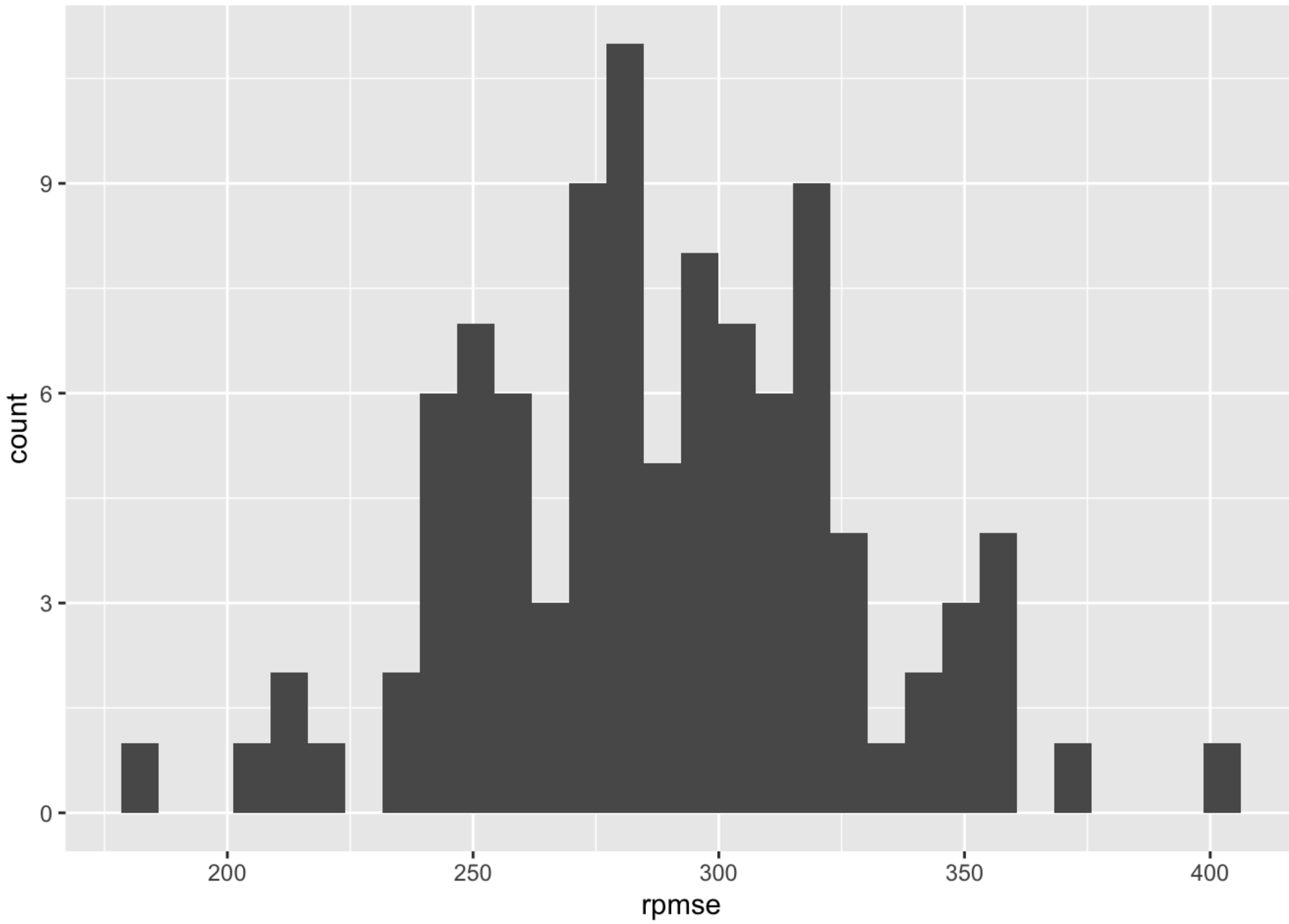
```
n.cv <- 100 #Number of CV studies to run
n.test <- 20  #Number of observations in a test set
rpmse <- rep(x=NA, times=n.cv)
bias <- rep(x=NA, times=n.cv)
wid <- rep(x=NA, times=n.cv)
cvg <- rep(x=NA, times=n.cv)
for(cv in 1:n.cv){
  ## Select test observations
  test.obs <- sample(x=1:n.cv, size=n.test)

  ## Split into test and training sets
  test.set <- birth_weights[test.obs,]
  train.set <- birth_weights[-test.obs,]

  ## Fit a lm() using the training data
  train.lm <- lm(formula=, data=train.set)

  ## Generate predictions for the test set
  my.preds <- predict.lm(train.lm, newdata=test.set, interval="prediction")

  ## Calculate bias
  bias[cv] <- mean(my.preds[,'fit']-test.set[['BirthWeight']])

  ## Calculate RPMSE
  rpmse[cv] <- (test.set[['BirthWeight']]-my.preds[,'fit'])^2 %>% mean() %>% sqrt()

  ## Calculate Coverage
  cvg[cv] <- ((test.set[['BirthWeight']] > my.preds[,'lwr']) & (test.set[['BirthWeigh
t']] < my.preds[,'upr'])) %>% mean()

  ## Calculate Width
  wid[cv] <- (my.preds[,'upr'] - my.preds[,'lwr']) %>% mean()

}

ggplot() + geom_histogram(mapping=aes(x=bias))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



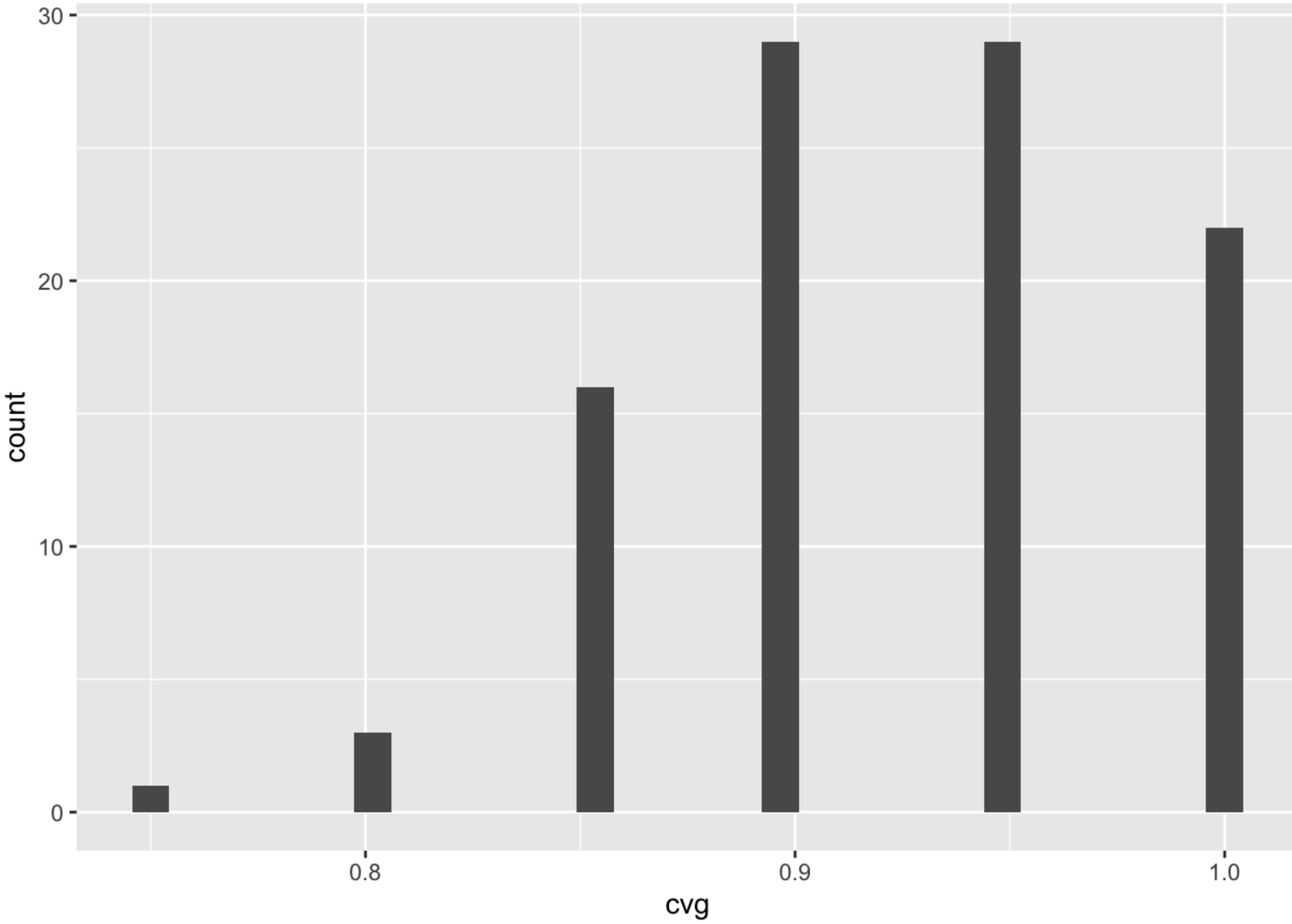

```
ggplot() + geom_histogram(mapping=aes(x=rpmse))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



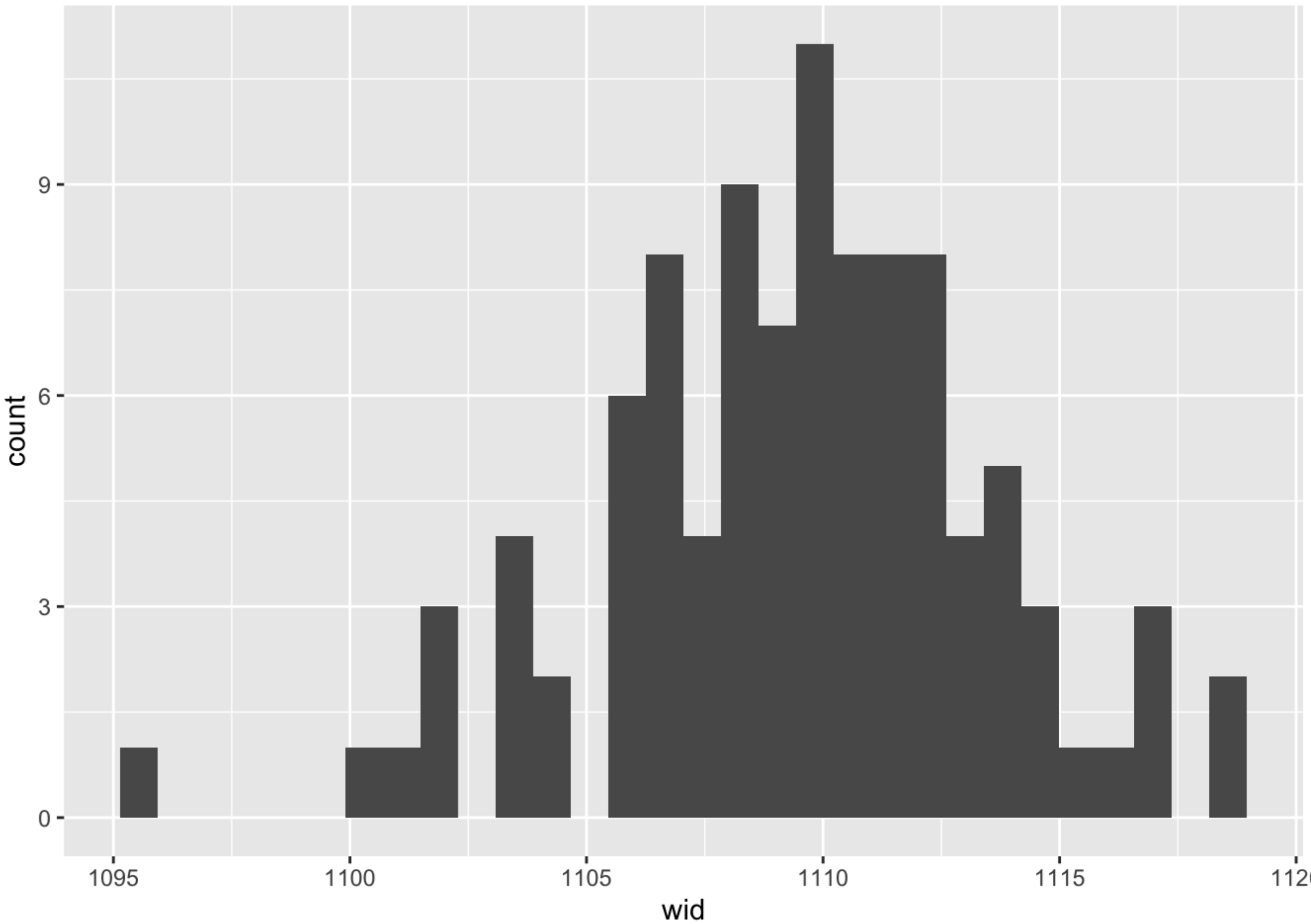

```
ggplot() + geom_histogram(mapping=aes(x=cvg))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggplot() + geom_histogram(mapping=aes(x=wid))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Hypothesis Testing and Confidence Intervals

1. Using lm() construct the t–statistic and p-value for the test H0:βMage=0.

```
mage.lm <- lm(formula=BirthWeight~Mage, data=birth_weights)
summary(mage.lm)
```

```
##
## Call:
## lm(formula = BirthWeight ~ Mage, data = birth_weights)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1505.86  -299.87     6.95   316.31  1605.17
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3212.938     83.297  38.572   <2e-16 ***
## Mage          -4.128      2.662  -1.551    0.121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 446.8 on 830 degrees of freedom
## Multiple R-squared:  0.002889,   Adjusted R-squared:  0.001687
## F-statistic: 2.404 on 1 and 830 DF,  p-value: 0.1214
```

2. Using confint() and lm(), build a 90% confidence interval for βMage.

```
confint(mage.lm,level=.90)
```

```
##                     5 %         95 %
## (Intercept) 3075.773315 3350.1029122
## Mage          -8.510954    0.2557274
```

3. Using anova(), conduct a Ftest that race has no effect on birth weight (note: this answers primary research question #2).

```
race.lm <- lm(formula=BirthWeight~Race, data=birth_weights)
anova(full.lm, race.lm)
```

```
## Analysis of Variance Table
##
## Model 1: BirthWeight ~ Mage + Gage + Race + Gen
## Model 2: BirthWeight ~ Race
##   Res.Df       RSS Df Sum of Sq      F    Pr(>F)
## 1    825  65403597
## 2    828 164278362 -3 -98874765 415.73 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. Using glht(), conduct a ttest and 94% confidence interval for the difference in average birth weight of babies born with explanatory variables

```
at <- t((c(1, 24, 40, 0, 0, 1, 1)-c(1, 34, 33, 0, 0, 1, 1)))

my.test <- glht(full.lm, linfct=at, alternative="two.sided")
summary(my.test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = BirthWeight ~ ., data = birth_weights)
##
## Linear Hypotheses:
##        Estimate Std. Error t value Pr(>|t|)
## 1 == 0  1317.13      40.48   32.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
confint(my.test,level=.94)
```

```
##
##   Simultaneous Confidence Intervals
##
## Fit: lm(formula = BirthWeight ~ ., data = birth_weights)
##
## Quantile = 1.8834
## 94% family-wise confidence level
##
##
## Linear Hypotheses:
##        Estimate  lwr       upr
## 1 == 0 1317.1350 1240.8954 1393.3746
```