In-Class Code Analysis #1

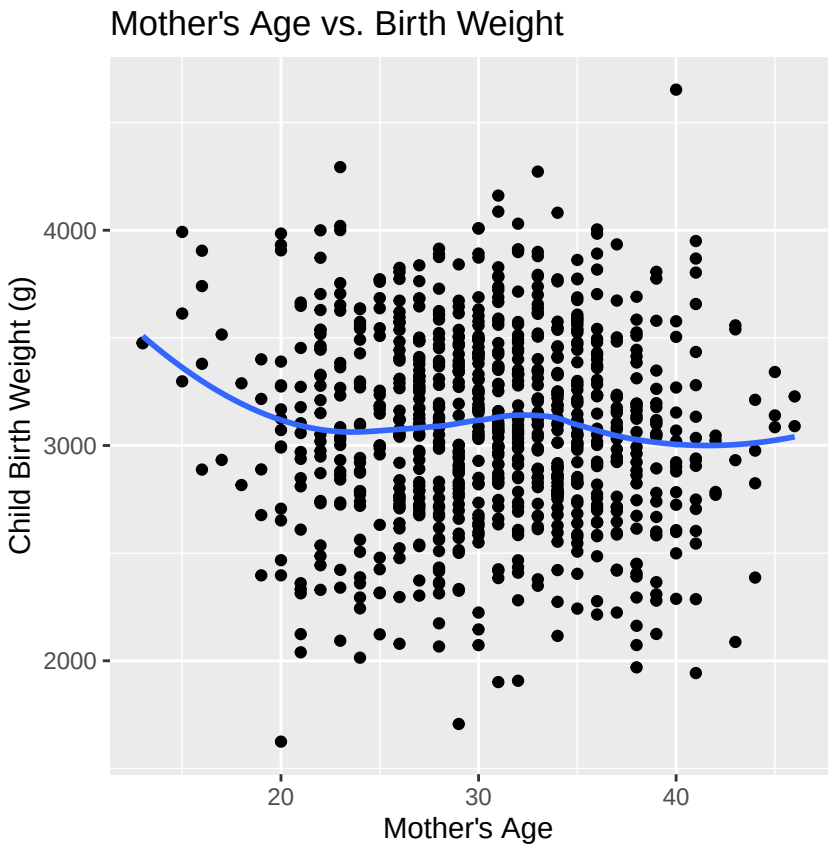
Jillian Warburton

2023-01-25

Exploratory Data Analysis

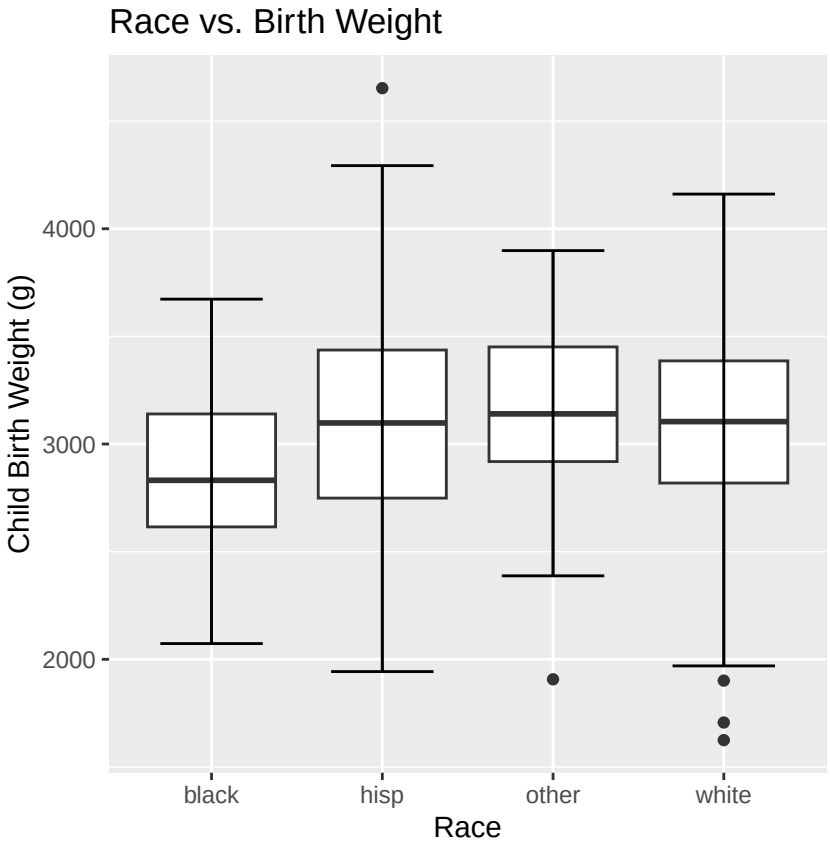
1. Scatterplot of `BirthWeight` by `Mage`

```
#scatterplot of BirthWeight vs Mage with trend line and axis labels
ggplot(data = data, mapping = aes(x = Mage, y = BirthWeight)) +
  geom_point() +
  theme(aspect.ratio = 1) +
  ggtitle("Mother's Age vs. Birth Weight") +
  xlab("Mother's Age") +
  ylab("Child Birth Weight (g)") +
  geom_smooth(se=FALSE)
```



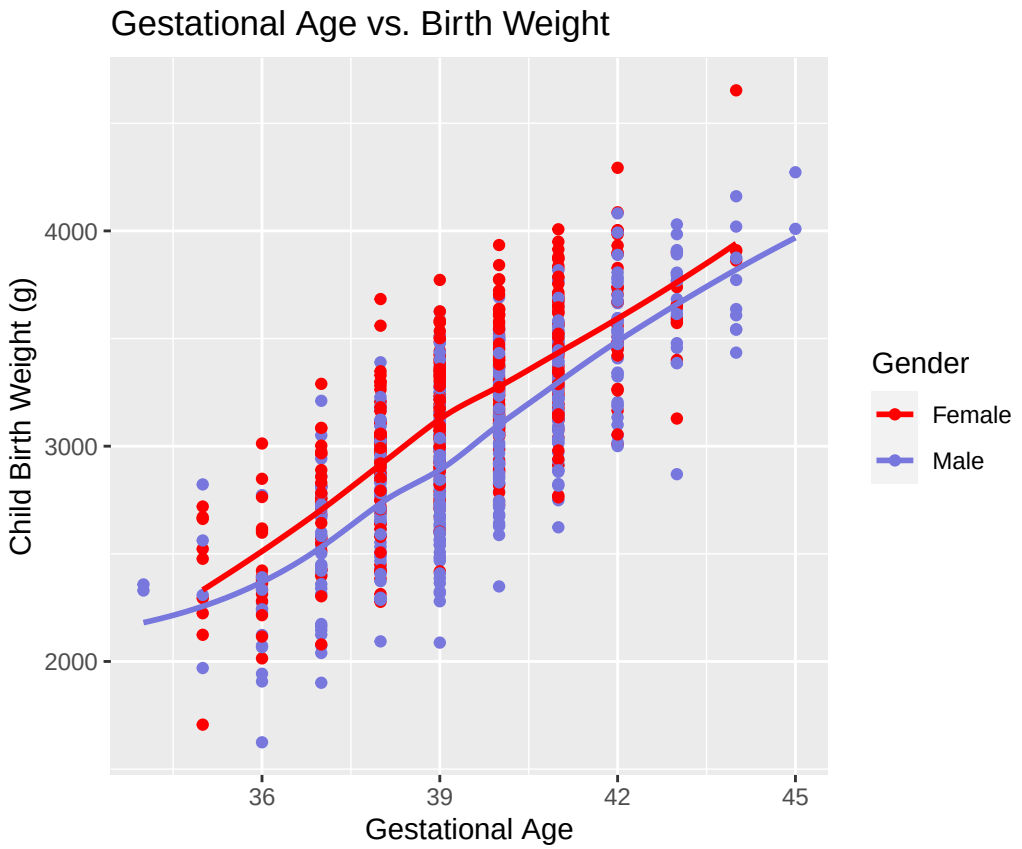
2. Side-by-side boxplots of `BirthWeight` for each category in `Race`

```
#boxplots of Race vs BirthWeight with axis labels and error bars
ggplot(data = data, mapping = aes(x = Race, y = BirthWeight)) +
  geom_boxplot() +
  theme(aspect.ratio = 1) +
  ggtitle("Race vs. Birth Weight") +
  xlab("Race") +
  ylab("Child Birth Weight (g)") +
  stat_boxplot(geom ='errorbar', width = 0.6)
```



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

```
#scatterplot of Gage vs BirthWeights, with Gen labels, axis labels, and trend line
ggplot(data = data, mapping = aes(x = Gage, y = BirthWeight, color = Gen)) +
  geom_point() +
  theme(aspect.ratio = 1) +
  ggtitle("Gestational Age vs. Birth Weight") +
  xlab("Gestational Age") +
  ylab("Child Birth Weight (g)") +
  scale_color_manual('Gender', values=c("red", "#7777DD")) +
  geom_smooth(se=FALSE)
```



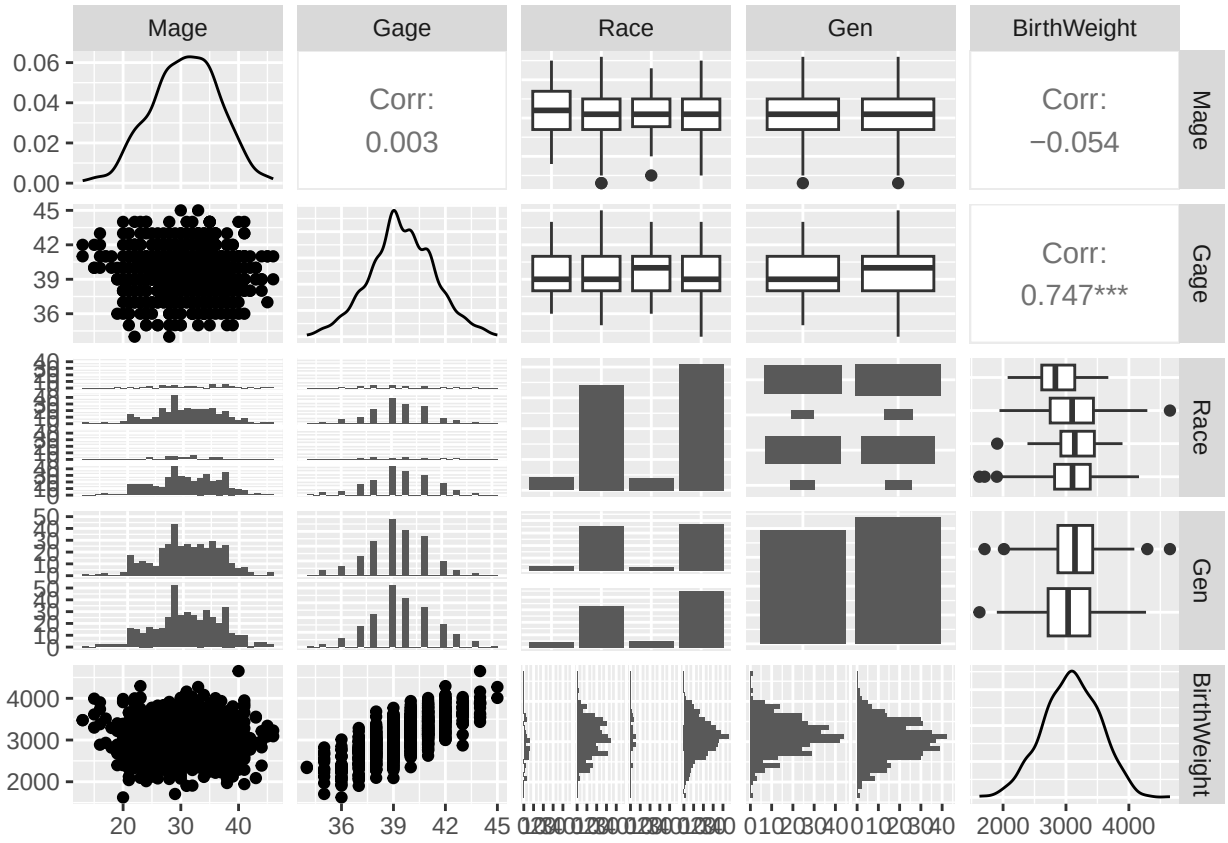
4. The correlation between `BirthWeight` and `Mage`.

```
#correlation of BirthWeight vs Mage
cor(data$Mage, data$BirthWeight)
```

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

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

```
ggpairs(data)
```



Fitting a Linear Model

1. Without the use of `lm()` calculate $\hat{\beta}$ and s^2. Verify your answer using `lm()`.

```
#calculate beta matrix, B^=((X'*X)^-1)*X'y
X <- model.matrix(object=BirthWeight~., data=data)
y_vect <- data$BirthWeight
beta.hat.mle <- (solve(t(X) %*% X)) %*% t(X) %*% y_vect
beta.hat.mle
```

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

```
#calculate std dev, s^2=(y-XB^)'(y-XB^)/(n-P-1)
P_birth=6
n_birth=nrow(data)
std.dev <- t(y_vect-X%*%beta.hat.mle) %*% (y_vect-X%*%beta.hat.mle) / (n_birth-P_birth-1)
std.dev
```

```
##          [,1]
## [1,] 79277.09
```

```
#Validate results
birth.lm <- lm(formula = BirthWeight~., data = data)
sigma(birth.lm)^2
```

```
## [1] 79277.09
```

```
summary(birth.lm)
```

```
##
## Call:
## lm(formula = BirthWeight ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -793.32 -196.79   -5.24  208.89  720.63
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -4120.542    218.050 -18.897 < 0.0000000000000002 ***
## Mage           -3.794      1.680  -2.259              0.024171 *
## Gage          182.742      5.256  34.770 < 0.0000000000000002 ***
## Racehisp      198.748     46.617   4.263             0.0000225 ***
## Raceother     241.583     62.639   3.857              0.000124 ***
## Racewhite     204.888     46.177   4.437             0.0000104 ***
## GenMale      -169.349     19.677  -8.607 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 281.6 on 825 degrees of freedom
## Multiple R-squared:  0.6065, Adjusted R-squared:  0.6036
## F-statistic: 211.9 on 6 and 825 DF,  p-value: < 0.00000000000000022
```

```
#ask if I should use above or below
#birth.stddev <- round(sigma(birth.lm), digits = 2)
```

2. Without the use of `lm()` calculate the fitted values $\mathbf{X}\hat{\beta}$. Verify your calculations by pulling off the fitted values from an `lm()` object.

```
#calculate fitted values
fit.mle <- X %*% beta.hat.mle

#compare to fitted values from birth.lm
summary(near(fit.mle, fitted(birth.lm)))
```

```
##     V1
##  Mode:logical
##  TRUE:832
```

3. Without the use of `lm()` calculate the residuals $\mathbf{y} - \mathbf{X}\hat{\beta}$. Verify your calculations by pulling off the residuals from an `lm()` object.

```
#calculate  residuals w/o lm()
resids.mle <- y_vect - X %*% beta.hat.mle

#compare to residuals from lm()
summary(near(resids.mle, resid(birth.lm)))
```

```
##     V1
##  Mode:logical
##  TRUE:832
```

4. Identify your model R^2 from the `summary()` output.

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

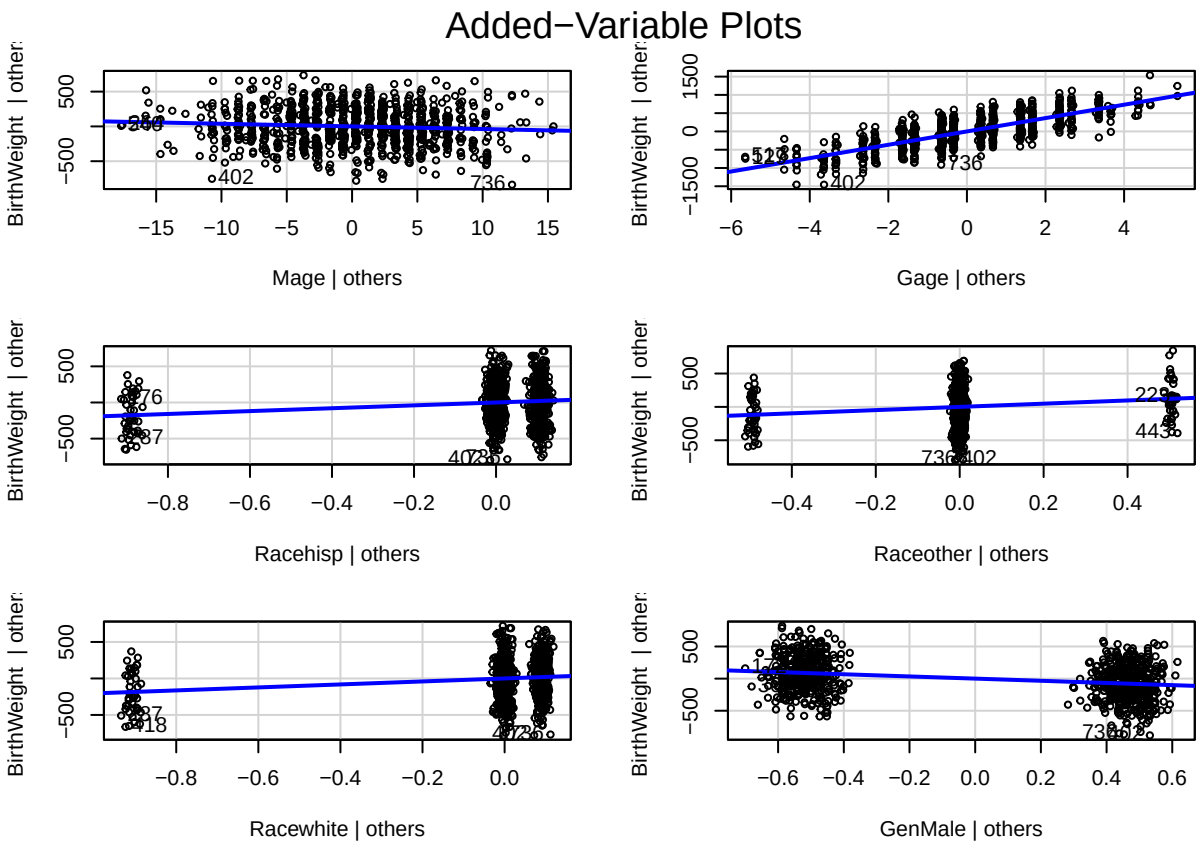
```
## [1] 0.6064689
```

```
#ask if I need
#birth.r2 <- 100 * round(summary(birth.lm)$r.squared, digits = 2)
```

Checking Assumptions

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

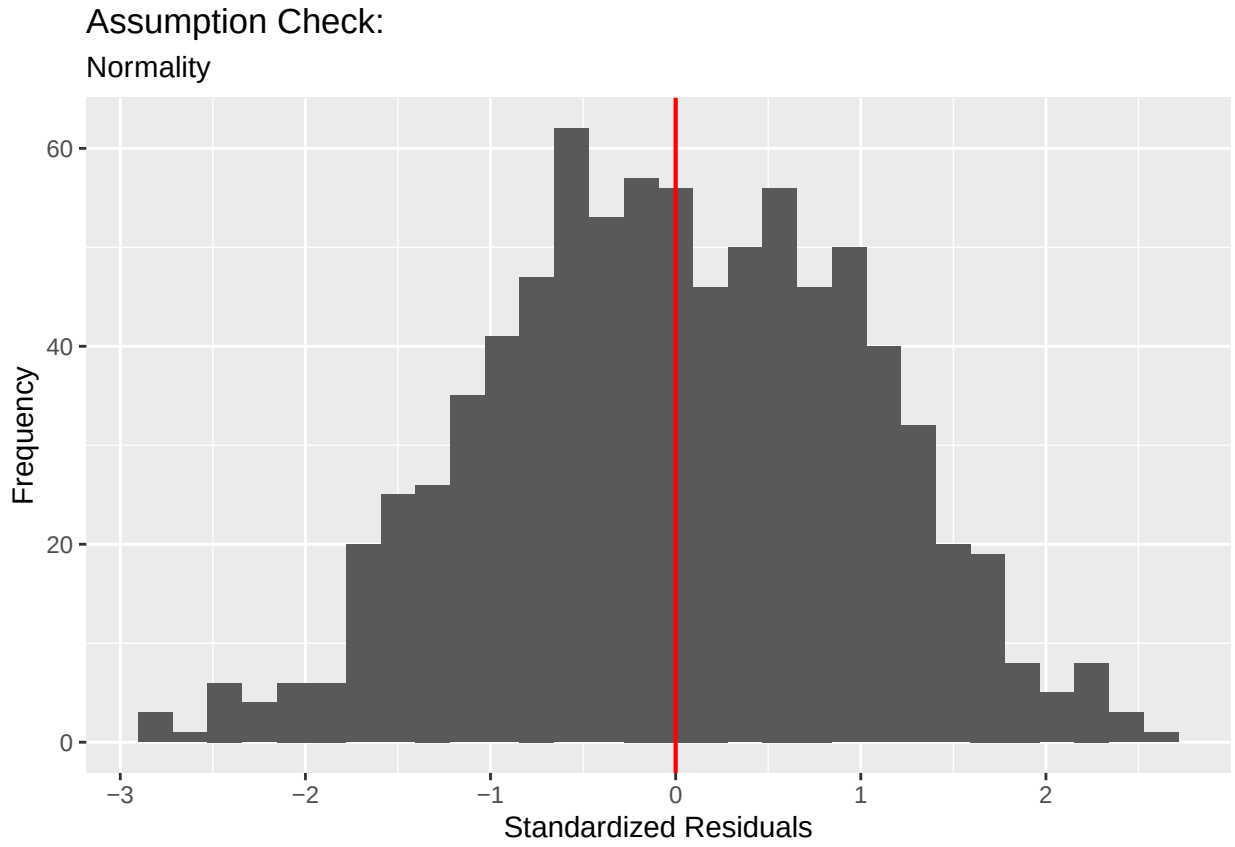
```
#make avPlots to check linearity
avPlots(birth.lm, ask = FALSE)
```



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

```
#check normality assumption with std res histogram
ggplot() +
  geom_histogram(mapping=aes(x=stdres(birth.lm))) +
  xlab('Standardized Residuals') +
  ylab('Frequency') +
  ggtitle('Assumption Check:', subtitle = 'Normality') +
  geom_vline(xintercept = 0, col = "red", lwd = 0.75)
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
```



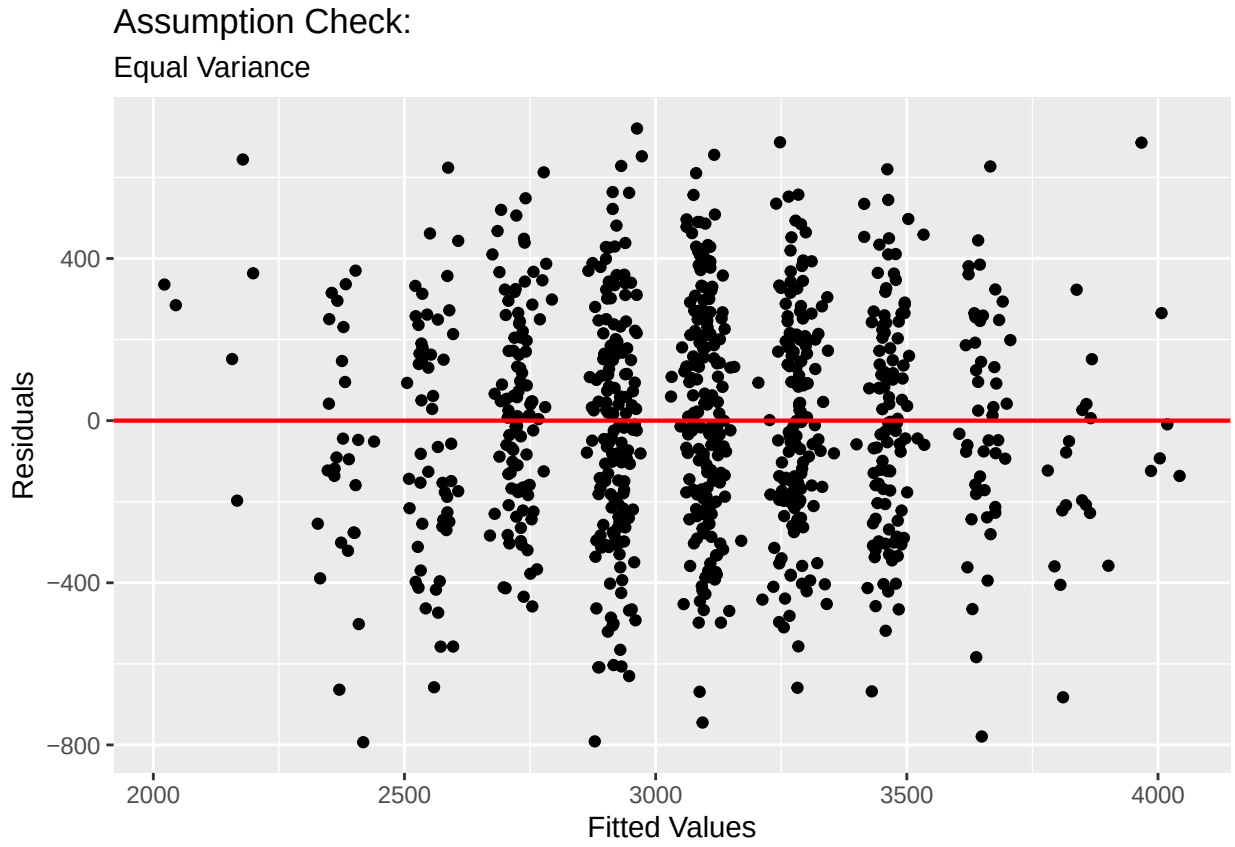

```
#check normality assumption with KS-test
ks.test(stdres(birth.lm), "pnorm")$p.value
```

```
## [1] 0.4884292
```

```
#ks.res <- round(ks.test(birth.stdres, "pnorm")$p.value, digits = 4)
```

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.

```
#check equal variance assumption with fit val vs std res scatterplot
ggplot(data, aes(x=birth.lm$fitted.values, y=birth.lm$residuals)) +
  geom_point() +
  xlab('Fitted Values') +
  ylab('Residuals') +
  ggtitle('Assumption Check:', subtitle ='Equal Variance') +
  geom_hline(yintercept = 0, col = "red", lwd = 0.75)
```




```
#check equal variance assumption with BP-test
bptest(birth.lm)$p.value
```

```
##        BP
## 0.3380368
```

```
#bp.res <- round(as.numeric(bptest(birth.lm)$p.value), digits = 4)
```

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 $\hat{y}_{new} = \mathbf{x}_{new}\hat{\beta}$ where $\hat{\beta}$ is the maximum likelihood estimate that you calculated above. Confirm that this is what `predict.lm()` is doing to get the point prediction.

```
#calculate BirthWeight of baby with Mage=26, Gage=37, Race="hisp", and Gen="Female"
baby <- c(1, 26, 37, 1, 0, 0 ,0)
baby %*% beta.hat.mle
```

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

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.

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

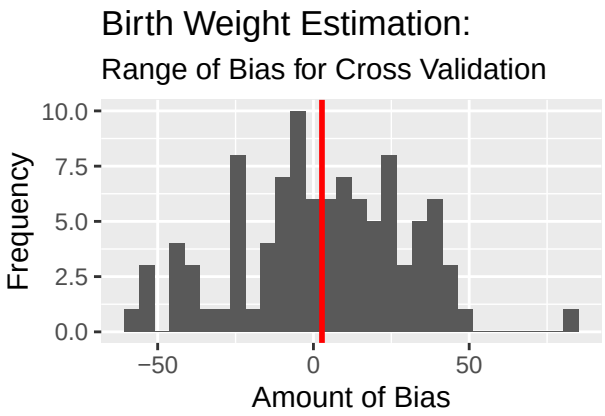
```
##       fit      lwr      upr
## 1 2741.04 2011.669 3470.412
```

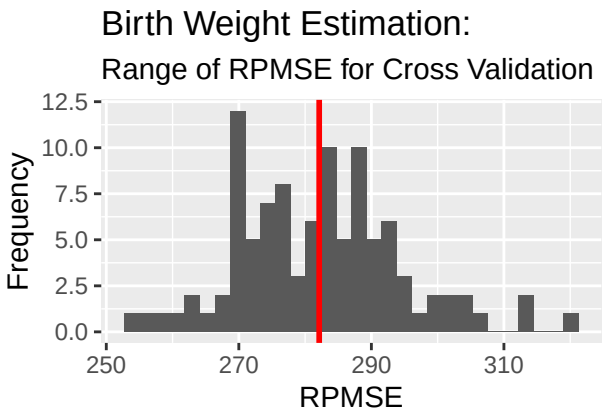
Cross Validation

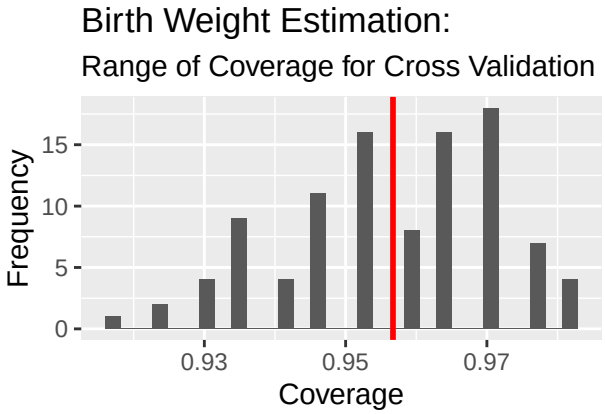
1. Adjust the code from class to run 100 Monte Carlo cross validations and plot histograms (or density plots) of the bias, RPMSE, coverage, and width.

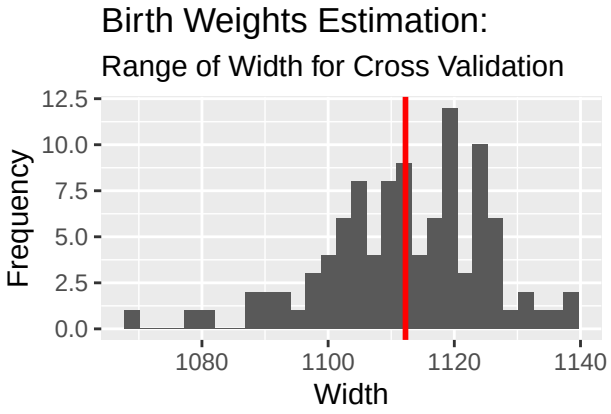
```r
set.seed(59) #for reproducibility
n.cv <- 100 #Number of CV studies to run
n.test <- 170 #Number of observations in a test set
# n.test = 170 is about 20% of 832
rpmse <- rep(x=NA, times=n.cv)
bias <- rep(x=NA, times=n.cv)
wid <- rep(x=NA, times=n.cv)
cvg <- rep(x=NA, times=n.cv)
for(cv in 1:n.cv){
  ## Select test observations
  test.obs <- sample(x=1:nrow(data), size=n.test)

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

  ## Fit a lm() using the training data
  train.lm <- lm(formula=BirthWeight~., 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[['BirthWeight']] < my.preds[,'upr'])) %>% mean()

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

}

CV.bias <- ggplot() +
  geom_histogram(mapping=aes(x=bias)) +
  xlab('Amount of Bias') +
  ylab('Frequency') +
```

```
  ggtitle('Birth Weight Estimation:',
          subtitle = 'Range of Bias for Cross Validation') +
  geom_vline(xintercept = mean(bias), col = "red", lwd = 1)

CV.RPMSE <- ggplot() +
  geom_histogram(mapping=aes(x=rpmse)) +
  xlab('RPMSE') +
  ylab('Frequency') +
  ggtitle('Birth Weight Estimation:',
          subtitle = 'Range of RPMSE for Cross Validation') +
  geom_vline(xintercept = mean(rpmse), col = "red", lwd = 1)

CV.coverage <- ggplot() +
  geom_histogram(mapping=aes(x=cvg)) +
  xlab('Coverage') +
  ylab('Frequency') +
  ggtitle('Birth Weight Estimation:',
          subtitle = 'Range of Coverage for Cross Validation') +
  geom_vline(xintercept = mean(cvg), col = "red", lwd = 1)

CV.width <- ggplot() +
  geom_histogram(mapping=aes(x=wid)) +
  xlab('Width') +
  ylab('Frequency') +
  ggtitle('Birth Weights Estimation:',
          subtitle = 'Range of Width for Cross Validation') +
  geom_vline(xintercept = mean(wid), col = "red", lwd = 1)

suppressMessages(grid.arrange(CV.bias, CV.RPMSE, CV.coverage, CV.width, nrow=2))
```









Hypothesis Testing and Confidence Intervals

1. Using `lm()` construct the $t-$statistic and $p-$value for the test $H_0 : \beta_{Mage} = 0$.

```
summary(birth.lm) #pull t-test and p-value
```

```
##
## Call:
## lm(formula = BirthWeight ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -793.32 -196.79   -5.24  208.89  720.63
##
## Coefficients:
##              Estimate Std. Error t value             Pr(>|t|)
## (Intercept) -4120.542    218.050 -18.897 < 0.0000000000000002 ***
## Mage           -3.794      1.680  -2.259              0.024171 *
## Gage          182.742      5.256  34.770 < 0.0000000000000002 ***
## Racehisp      198.748     46.617   4.263             0.0000225 ***
## Raceother     241.583     62.639   3.857              0.000124 ***
## Racewhite     204.888     46.177   4.437             0.0000104 ***
## GenMale      -169.349     19.677  -8.607 < 0.0000000000000002 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 281.6 on 825 degrees of freedom
## Multiple R-squared:  0.6065, Adjusted R-squared:  0.6036
## F-statistic: 211.9 on 6 and 825 DF,  p-value: < 0.00000000000000022
```

t-statistic for $H_0 : \beta_{Mage} = 0$: -2.259 p-value for $H_0 : \beta_{Mage} = 0$: 0.024171

2. Using `confint()` and `lm()`, build a 90% confidence interval for β_{Mage}.

```
betas.ci <- confint(birth.lm, level = 0.9)
betas.ci[2,1]
```

```
## [1] -6.559754
```

```
betas.ci[2,2]
```

```
## [1] -1.027749
```

3. Using `anova()`, conduct a $F-$test that race has no effect on birth weight *(note: this answers primary research question #2).*

```
reduced.lm <- lm(BirthWeight~.-Race, data)
anova(birth.lm,reduced.lm)
```

```
## Analysis of Variance Table
##
## Model 1: BirthWeight ~ Mage + Gage + Race + Gen
## Model 2: BirthWeight ~ (Mage + Gage + Race + Gen) - Race
##   Res.Df      RSS Df Sum of Sq      F   Pr(>F)
## 1    825 65403597
## 2    828 67089176 -3  -1685579 7.0873 0.000105 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4. Using `glht()`, conduct a $t-$test and 94% confidence interval for the difference in average birth weight of babies born with explanatory variables `Mage=24`, `Gage=40`, `Race="white"`, and `Gen="Male"` and babies born with explanatory variables `Mage=34`, `Gage=33`, `Race="white"`, and `Gen="Male"`.

```
baby_1 <- c(1, 24, 40, 0, 0, 1, 1)
baby_2 <- c(1, 34, 33, 0, 0, 1, 1)
my.test <- glht(birth.lm,
                linfct=t(baby_1 - baby_2),
                alternative="two.sided")
summary(my.test)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lm(formula = BirthWeight ~ ., data = data)
```

```
##
## Linear Hypotheses:
##        Estimate Std. Error t value            Pr(>|t|)
## 1 == 0  1317.13      40.48   32.54 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
confint(my.test, 0.94)
```

```
##
##   Simultaneous Confidence Intervals
##
## Fit: lm(formula = BirthWeight ~ ., data = data)
##
## Quantile = 1.9628
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##        Estimate  lwr       upr
## 1 == 0 1317.1350 1237.6788 1396.5912
```