DA5020.A10.Hsiao-Yu.Peng

Hsiao-Yu Peng

2023-11-30

Q1-1.In your own words, provide a clear definition of the confidence interval and the prediction interval, and state their respective significance.

1. *Confidence Interval (CI):*

- Definition: A confidence interval is a statistical range that is calculated from sample data and is used to estimate the range within which a population parameter, such as the mean or proportion, is likely to fall. It provides a level of confidence, typically expressed as a percentage (e.g., 95% confidence interval), that the true parameter value is within the calculated interval.
- Significance: The confidence interval gives us a sense of the precision of our estimate. For example, a 95% confidence interval means that if we were to take many samples and construct confidence intervals from each, we would expect about 95% of those intervals to contain the true parameter.

2. *Prediction Interval (PI):*

- Definition: A prediction interval is also a statistical range, but it is used to estimate the range within which a future observation or data point is likely to fall. Unlike a confidence interval, which focuses on estimating a population parameter, a prediction interval takes into account both the variability of the data and the uncertainty associated with predicting an individual outcome.
- Significance: The prediction interval is broader than the confidence interval because it considers not only the variability in estimating the population parameter but also the variability of individual observations. It accounts for both the variability within the sample data and the uncertainty associated with predicting an individual value.

Q1-2. Describe in your own words what a multiple linear regression is and why one would be used.

Multiple linear regression analyzes the relationship between multiple independent variables and a single dependent variable. In simpler terms, it helps us understand how several factors influence or contribute to the variation in an outcome.

Multiple linear regression is employed when the relationship between the dependent variable and multiple independent variables is more intricate than what can be captured by a simple linear model. By considering multiple factors simultaneously, the model aims to improve the accuracy of predictions compared to models with fewer variables.

It helps identify which independent variables are statistically significant predictors of the dependent variable, aiding in understanding the relative importance of each factor.

Q1-3. Install the openintro R package and load the library in your R environment. Use the ncbirths dataset to answer the following questions

```
## install.packages("openintro")
# "ncbirths" not found, use `install.packages("Stat2Data")` instead.

df <- ncbirths

summary(df)
```

```
##       fage            mage            mature        weeks            premie
##  Min.   :14.00   Min.   :13   mature mom :133   Min.   :20.00   full term:846
##  1st Qu.:25.00   1st Qu.:22   younger mom:867   1st Qu.:37.00   premie   :152
##  Median :30.00   Median :27                     Median :39.00   NA's     :  2
##  Mean   :30.26   Mean   :27                     Mean   :38.33
##  3rd Qu.:35.00   3rd Qu.:32                     3rd Qu.:40.00
##  Max.   :55.00   Max.   :50                     Max.   :45.00
##  NA's   :171                                    NA's   :2
##      visits           marital        gained          weight
##  Min.   : 0.0   not married:386   Min.   : 0.00   Min.   : 1.000
##  1st Qu.:10.0   married    :613   1st Qu.:20.00   1st Qu.: 6.380
##  Median :12.0   NA's       :  1   Median :30.00   Median : 7.310
##  Mean   :12.1                     Mean   :30.33   Mean   : 7.101
##  3rd Qu.:15.0                     3rd Qu.:38.00   3rd Qu.: 8.060
##  Max.   :30.0                     Max.   :85.00   Max.   :11.750
##  NA's   :9                        NA's   :27
##  lowbirthweight    gender          habit          whitemom
##  low    :111    female:503   nonsmoker:873   not white:284
##  not low:889    male  :497   smoker   :126   white    :714
##                              NA's     :  1   NA's     :  2
##
##
##
##
```

```
glimpse(df)
```

```
## Rows: 1,000
## Columns: 13
## $ fage           <int> NA, NA, 19, 21, NA, NA, 18, 17, NA, 20, 30, NA, NA, NA,~
## $ mage           <int> 13, 14, 15, 15, 15, 15, 15, 15, 16, 16, 16, 16, 16, 16,~
## $ mature         <fct> younger mom, younger mom, younger mom, younger mom, you~
## $ weeks          <int> 39, 42, 37, 41, 39, 38, 37, 35, 38, 37, 45, 42, 40, 38,~
## $ premie         <fct> full term, full term, full term, full term, full term, ~
## $ visits         <int> 10, 15, 11, 6, 9, 19, 12, 5, 9, 13, 9, 8, 4, 12, 15, 7,~
## $ marital        <fct> not married, not married, not married, not married, not~
## $ gained         <int> 38, 20, 38, 34, 27, 22, 76, 15, NA, 52, 28, 34, 12, 30,~
## $ weight         <dbl> 7.63, 7.88, 6.63, 8.00, 6.38, 5.38, 8.44, 4.69, 8.81, 6~
## $ lowbirthweight <fct> not low, not low, not low, not low, not low, low, not l~
```

```
## $ gender         <fct> male, male, female, male, female, male, male, male, mal~
## $ habit          <fct> nonsmoker, nonsmoker, nonsmoker, nonsmoker, nonsmoker, ~
## $ whitemom       <fct> not white, not white, white, white, not white, not whit~
```

The data set has dimension 1000 x 13, containing both numeric and factor data types. We will preprocess factor variables using either label encoding or one-hot encoding.

Find out what variables has NA values

```
# Check for missing values in each variable
colSums(is.na(df))
```

```
##           fage           mage         mature          weeks         premie
##            171              0              0              2              2
##         visits        marital         gained         weight lowbirthweight
##              9              1             27              0              0
##         gender          habit       whitemom
##              0              1              2
```

Because the data set has some NA values, we will perform data imputation.

```
# Impute values in fage, weeks, visits, gained
df <- df %>%
  mutate(fage = ifelse(is.na(fage), mean(fage, na.rm = TRUE), fage)) %>%
  mutate(weeks = ifelse(is.na(weeks), median(weeks, na.rm = TRUE), weeks)) %>%
  mutate(visits = ifelse(is.na(visits), median(visits, na.rm = TRUE), visits)) %>%
  mutate(gained = ifelse(is.na(gained), median(gained, na.rm = TRUE), gained))
```

Numeric Variables Preprocessing The remaining variables with NA values are factor variables, which are quite few. Therefore, we can ignore them for the following calculation.

```
# Remove rows with NA values
df <- na.omit(df)
colSums(is.na(df))
```

```
##           fage           mage         mature          weeks         premie
##              0              0              0              0              0
##         visits        marital         gained         weight lowbirthweight
##              0              0              0              0              0
##         gender          habit       whitemom
##              0              0              0
```

```
dim(df)
```

```
## [1] 996  13
```

Now we don't have NA values in the data set.

```r
# Select all factor variables from the dataset
factor_var <- df %>%
  select_if(~ !is.numeric(.))

names(factor_var)
```

Factor Variables PreProcessing

```
## [1] "mature"         "premie"         "marital"        "lowbirthweight"
## [5] "gender"         "habit"          "whitemom"
```

```r
# Identify unique character values in each factor variable
unique_values <- lapply(factor_var, levels)

unique_values
```

```
## $mature
## [1] "mature mom"  "younger mom"
##
## $premie
## [1] "full term" "premie"
##
## $marital
## [1] "not married" "married"
##
## $lowbirthweight
## [1] "low"     "not low"
##
## $gender
## [1] "female" "male"
##
## $habit
## [1] "nonsmoker" "smoker"
##
## $whitemom
## [1] "not white" "white"
```

From the level results of factor variables, we can see that they have a binary definition. Therefore, we will apply label encoding, assigning 0 and 1 to these factor variables.

```r
# Label Encoding
transformed_df <- df %>%
  mutate(mature = if_else(mature == unique_values$mature[1], 1, 0)) %>%
  mutate(premie = if_else(premie == unique_values$premie[2], 1, 0)) %>%
  mutate(marital = if_else(marital == unique_values$marital[2], 1, 0)) %>%
  mutate(lowbirthweight = if_else(lowbirthweight == unique_values$lowbirthweight[1], 1, 0)) %>%
  mutate(gender = if_else(gender == unique_values$gender[1], 1, 0)) %>%
  mutate(habit = if_else(habit == unique_values$habit[2], 1, 0)) %>%
  mutate(whitemom = if_else(whitemom == unique_values$whitemom[2], 1, 0))

head(transformed_df)
```

```
## # A tibble: 6 x 13
##    fage  mage mature weeks premie visits marital gained weight lowbirthweight
##   <dbl> <int>  <dbl> <dbl>  <dbl>  <int>   <dbl>  <int>  <dbl>          <dbl>
## 1  30.3    13      0    39      0     10       0     38   7.63              0
## 2  30.3    14      0    42      0     15       0     20   7.88              0
## 3  19      15      0    37      0     11       0     38   6.63              0
## 4  21      15      0    41      0      6       0     34   8                 0
## 5  30.3    15      0    39      0      9       0     27   6.38              0
## 6  30.3    15      0    38      0     19       0     22   5.38              1
## # i 3 more variables: gender <dbl>, habit <dbl>, whitemom <dbl>
```

Q2. Load the data in your R environment and build a full correlation matrix ,i.e. a matrix that shows the correlations between all variables. Do you detect any multicollinearity that would affect the construction of a multiple regression model? Comment on the distribution of each field. Do you anticipate that there are fields that may not be useful for the model? If yes, provide an example.

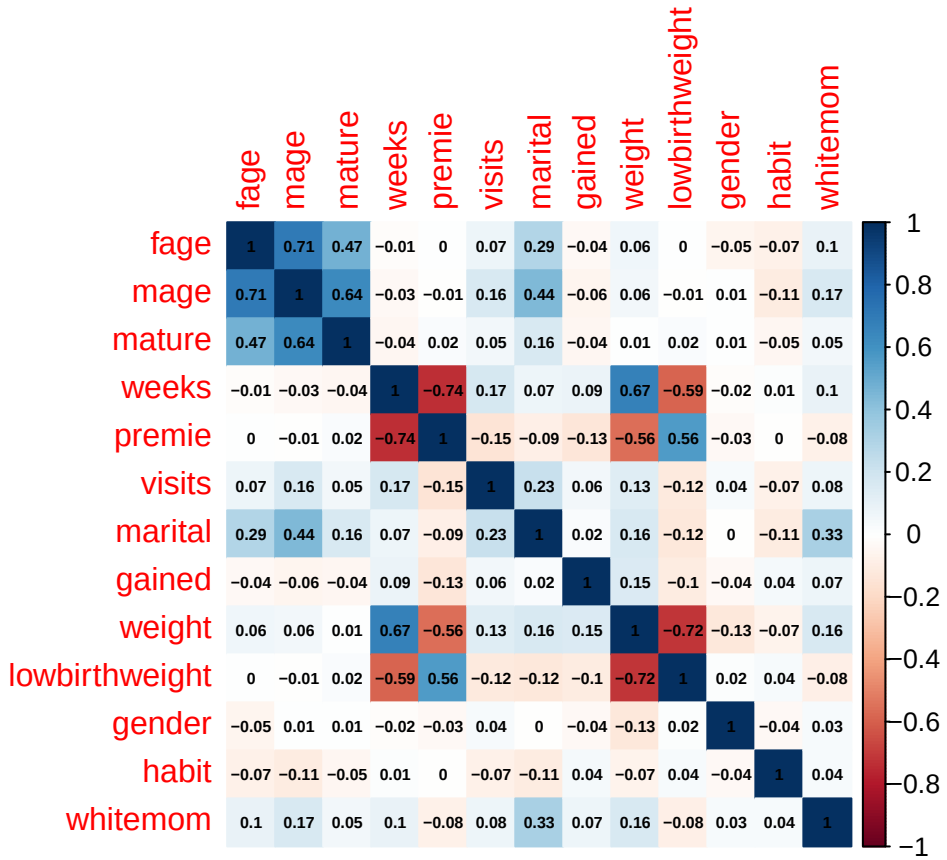
```
# Calculate correlation matrix
cor_matrix <- cor(transformed_df, use = "pairwise.complete.obs") # ignore NA values

print(cor_matrix)
```

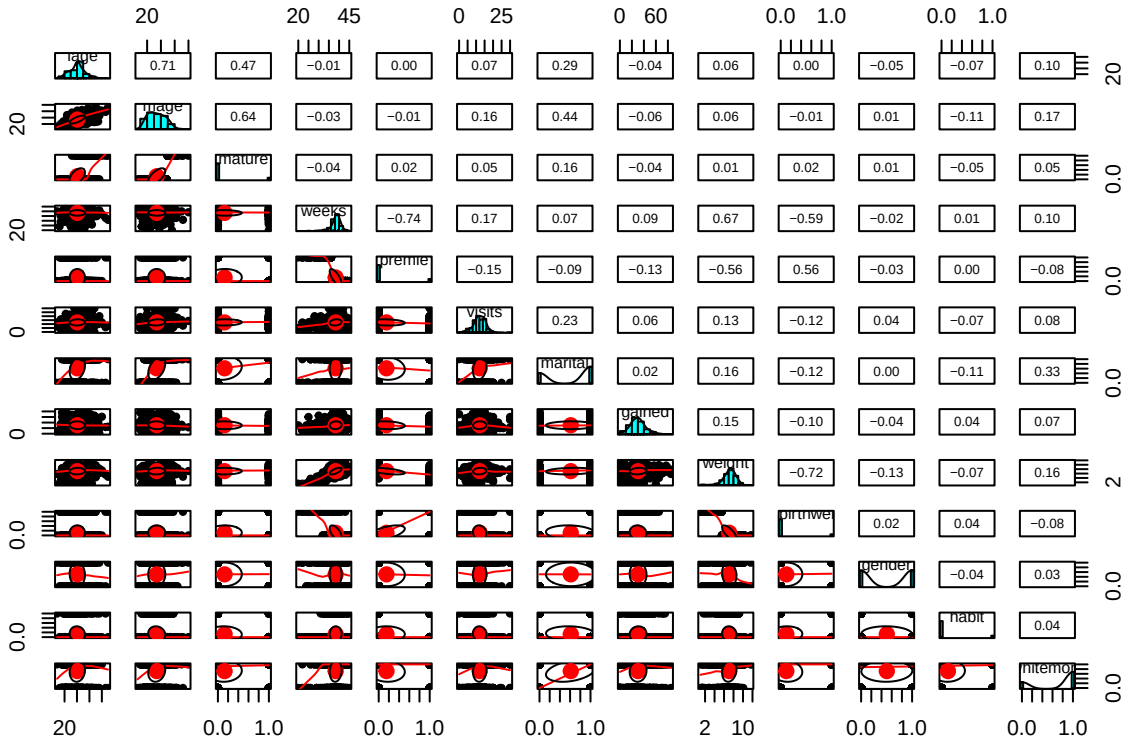
```
##                       fage         mage       mature       weeks       premie
## fage            1.000000000  0.708064183  0.474958685 -0.01471788  0.002981144
## mage            0.708064183  1.000000000  0.638618429 -0.03398831 -0.006572980
## mature          0.474958685  0.638618429  1.000000000 -0.04398030  0.024848781
## weeks          -0.014717878 -0.033988311 -0.043980299  1.00000000 -0.735535345
## premie          0.002981144 -0.006572980  0.024848781 -0.73553535  1.000000000
## visits          0.071383449  0.163263848  0.052140714  0.17288690 -0.146183064
## marital         0.294905689  0.442560998  0.162502089  0.07001820 -0.087406194
## gained         -0.037740698 -0.059039468 -0.041566024  0.08953379 -0.130504023
## weight          0.060989777  0.056538769  0.007361037  0.67023461 -0.557056719
## lowbirthweight  0.002127291 -0.007656560  0.023998725 -0.58940060  0.563064956
## gender         -0.054213870  0.008958194  0.005786457 -0.01803159 -0.031329109
## habit          -0.074021774 -0.106033997 -0.049797138  0.01468378 -0.001922635
## whitemom        0.098233200  0.167302955  0.054964269  0.09792439 -0.078280577
##                     visits      marital      gained       weight lowbirthweight
## fage            0.07138345  0.294905689 -0.03774070  0.060989777    0.002127291
## mage            0.16326385  0.442560998 -0.05903947  0.056538769   -0.007656560
## mature          0.05214071  0.162502089 -0.04156602  0.007361037    0.023998725
## weeks           0.17288690  0.070018201  0.08953379  0.670234605   -0.589400604
## premie         -0.14618306 -0.087406194 -0.13050402 -0.557056719    0.563064956
## visits          1.00000000  0.228569744  0.05832557  0.134926721   -0.115863635
## marital         0.22856974  1.000000000  0.02475915  0.163641670   -0.121561274
## gained          0.05832557  0.024759148  1.00000000  0.151670250   -0.100904349
## weight          0.13492672  0.163641670  0.15167025  1.000000000   -0.717856772
## lowbirthweight -0.11586363 -0.121561274 -0.10090435 -0.717856772    1.000000000
## gender          0.03774801 -0.003933302 -0.03587062 -0.129598282    0.016389696
```

```
## habit           -0.07460246 -0.107272543  0.04181969 -0.069337115    0.039357077
## whitemom         0.07690440  0.329816212  0.07235341  0.160917043   -0.082546016
##                      gender        habit    whitemom
## fage            -0.054213870 -0.074021774  0.09823320
## mage             0.008958194 -0.106033997  0.16730296
## mature           0.005786457 -0.049797138  0.05496427
## weeks           -0.018031589  0.014683775  0.09792439
## premie          -0.031329109 -0.001922635 -0.07828058
## visits           0.037748008 -0.074602455  0.07690440
## marital         -0.003933302 -0.107272543  0.32981621
## gained          -0.035870618  0.041819694  0.07235341
## weight          -0.129598282 -0.069337115  0.16091704
## lowbirthweight   0.016389696  0.039357077 -0.08254602
## gender           1.000000000 -0.039302011  0.03175976
## habit           -0.039302011  1.000000000  0.03965540
## whitemom         0.031759763  0.039655395  1.00000000
```

```r
# Display the correlation matrix plot
corrplot(cor_matrix, method = "color", addCoef.col = "black", number.cex = 0.5)
```




```r
# Display the correlation coefficienct, scatter plot, and variables histograms
pairs.panels(transformed_df)
```



Multicollinearity happens when independent variables in the regression model are highly correlated to each other. We observed that "fage" appears to be related to "mage" (r = 0.71), "premie" and "weeks" show a correlation (r = -0.74). Finally, "lowbirthweight" and "weight" seem to be related (r = -0.72). The high correlation among some of the predictors suggests that data-based multicollinearity exists.

The variables—fage, mage, visits, gained, and weight—exhibit somewhat normal distributions, while weeks displays a left-skewed distribution. On the other hand, other variables—mature, premie, marital, lowbirthweight, gender, habit, and whitemom—have been transformed into binary definitions. The "Mature" variable indicates that younger moms outnumber mature moms, the "Premie" variable suggests that full-term cases are more common than preterm, "Marital" indicates a slightly higher frequency of married compared to unmarried, "Lowbirthweight" suggests that the number of infants with normal birth weight is higher than those with low birth weight, gender distribution is nearly equal, and the smoking habit variable shows that non-smokers outnumber smokers.

As we will build a model to predict weight later, we assess correlation coefficients related to weight. We find that "fage," "mage," "mature," "visits," and "habit" have lower coefficients ($r < 0.1$) in relation to weight. These variables may not be particularly useful for the model. However, we will confirm this by constructing an initial regression model with all variables and then use backward elimination to evaluate the useful variables.

Q3. Build a full multiple regression model that predicts the birth weight i.e weight. Comment on the: R-squared, Standard Error, F-Statistic, p-values of coefficients.

```
# Fit a linear model, assuming 'weight' is the response variable
model <- lm(weight ~ ., data = transformed_df)

# Print a summary of the regression model
summary(model)
```

```
##
## Call:
## lm(formula = weight ~ ., data = transformed_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4702 -0.6058 -0.0191  0.5568  3.9838
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.542248   0.642157  -0.844 0.398642
## fage            0.007614   0.006654   1.144 0.252787
## mage            0.003789   0.008244   0.460 0.645889
## mature          0.009361   0.112399   0.083 0.933640
## weeks           0.192148   0.015421  12.460  < 2e-16 ***
## premie         -0.004511   0.122336  -0.037 0.970595
## visits         -0.002622   0.007666  -0.342 0.732410
## marital         0.102047   0.071134   1.435 0.151725
## gained          0.006942   0.002078   3.341 0.000867 ***
## lowbirthweight -2.287723   0.117575 -19.458  < 2e-16 ***
## gender         -0.347002   0.058002  -5.983 3.07e-09 ***
## habit          -0.266957   0.087788  -3.041 0.002421 **
## whitemom        0.233046   0.068114   3.421 0.000649 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9068 on 983 degrees of freedom
## Multiple R-squared:  0.6413, Adjusted R-squared:  0.6369
## F-statistic: 146.4 on 12 and 983 DF,  p-value: < 2.2e-16
```

1. **R-squared**: The R-squared value is 0.6413, which means that approximately 64.13% of the variance in the birth weight can be explained by the predictor variables in the model. This indicates a moderately good fit.
2. **Adjusted R-squared**: The Adjusted R-squared value is 0.6369. This adjusts the R-squared value for the number of predictors in the model, providing a more accurate measure of model fit, especially when adding more predictors.
3. **Residual Standard Error**: The residual standard error (0.9068) represents the standard deviation of the residuals. It gives an estimate of the average amount by which the actual birth weight values deviate from the predicted values.
4. **F-Statistic**: The F-statistic (146.4) tests the overall significance of the regression model. In this case, the large F-statistic with a very low p-value ($< 2.2e\text{-}16$) suggests that the model as a whole is statistically significant.

5. **p-values of Coefficients**:

- Variables with low p-values < 0.05 (e.g., 'weeks,' 'gained,' 'lowbirthweight,' 'gender,' 'habit,' and 'whitemom') are likely to be significant predictors of birth weight.
- 'mature,' 'premie,' 'visits,' 'marital,' 'fage,' and 'mage' have higher p-values > 0.05 and are not considered statistically significant predictors.

Q4. Build a multiple regression model in which all coefficients are significant — use stepwise elimination based on coefficients with the p-value > 0.05. Show each step as you eliminate the coefficients and clearly state the reason for their elimination. At each step, determine if the model is improving.

Arrange the coefficients with p value > 0.05 from high to low.

```
# Extract p-values and variable names
sm <- summary(model)
p_values <- sm$coefficients[, "Pr(>|t|)"]
variable_names <- rownames(sm$coefficients)

# Create a data frame with variable names and p-values
result_df <- data.frame(variable = variable_names, p_value = p_values)

# Order the data frame by p-values in descending order
result_df <- result_df[order(result_df$p_value, decreasing = TRUE), ]

# Print the ordered result
print(result_df)
```

```
##                       variable      p_value
## premie                 premie 9.705947e-01
## mature                 mature 9.336405e-01
## visits                 visits 7.324098e-01
## mage                     mage 6.458890e-01
## (Intercept)       (Intercept) 3.986421e-01
## fage                     fage 2.527868e-01
## marital               marital 1.517246e-01
## habit                   habit 2.420937e-03
## gained                 gained 8.666255e-04
## whitemom             whitemom 6.485873e-04
## gender                 gender 3.073227e-09
## weeks                   weeks 3.418875e-33
## lowbirthweight lowbirthweight 1.372127e-71
```

p value > 0.05 in a descending order: 'premie,' 'mature,' 'visits' 'mage,' 'fage,' and 'marital'

```
# Stepwise elimination
stepwise_model <- model
```

```r
# remove "premie" variable
transformed_df2 <- subset(transformed_df, select = -c(premie))

stepwise_model <- lm(weight ~., data = transformed_df2)
cat("Removed: 'premie'", "\n")
```

```
## Removed: 'premie'
```

```r
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = weight ~ ., data = transformed_df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4715 -0.6058 -0.0187  0.5563  3.9841
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.556429   0.513998  -1.083 0.279273
## fage            0.007610   0.006650   1.144 0.252752
## mage            0.003802   0.008232   0.462 0.644314
## mature          0.009292   0.112326   0.083 0.934085
## weeks           0.192490   0.012313  15.633  < 2e-16 ***
## visits         -0.002620   0.007661  -0.342 0.732479
## marital         0.102077   0.071093   1.436 0.151368
## gained          0.006949   0.002069   3.359 0.000812 ***
## lowbirthweight -2.288719   0.114377 -20.010  < 2e-16 ***
## gender         -0.346850   0.057826  -5.998  2.8e-09 ***
## habit          -0.266933   0.087741  -3.042 0.002410 **
## whitemom        0.232999   0.068067   3.423 0.000645 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9063 on 984 degrees of freedom
## Multiple R-squared:  0.6413, Adjusted R-squared:  0.6373
## F-statistic: 159.9 on 11 and 984 DF,  p-value: < 2.2e-16
```

We eliminated the "premie" variable, and the R-squared and standard error did not change significantly after fitting the model.

```r
# remove "mature" variable
transformed_df2 <- subset(transformed_df2, select = -c(mature))

stepwise_model <- lm(weight ~., data = transformed_df2)
cat("Removed: 'mature'", "\n")
```

```
## Removed: 'mature'
```

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = weight ~ ., data = transformed_df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4711 -0.6029 -0.0187  0.5548  3.9813
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.564283   0.504899  -1.118 0.264003
## fage            0.007629   0.006642   1.149 0.251028
## mage            0.004147   0.007091   0.585 0.558769
## weeks           0.192490   0.012307  15.641  < 2e-16 ***
## visits         -0.002642   0.007653  -0.345 0.729978
## marital         0.101197   0.070255   1.440 0.150070
## gained          0.006951   0.002068   3.362 0.000804 ***
## lowbirthweight -2.288637   0.114315 -20.021  < 2e-16 ***
## gender         -0.346830   0.057797  -6.001 2.75e-09 ***
## habit          -0.266852   0.087691  -3.043 0.002404 **
## whitemom        0.232890   0.068020   3.424 0.000643 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9059 on 985 degrees of freedom
## Multiple R-squared:  0.6413, Adjusted R-squared:  0.6376
## F-statistic: 176.1 on 10 and 985 DF,  p-value: < 2.2e-16
```

Next, we eliminated the "mature" variable, and the standard error and R-squared did not change significantly after fitting the model.

```
# remove "visits" variable
transformed_df2 <- subset(transformed_df2, select = -c(visits))

stepwise_model <- lm(weight ~., data = transformed_df2)
cat("Removed: 'visits'", "\n")
```

```
## Removed: 'visits'
```

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = weight ~ ., data = transformed_df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4827 -0.6017 -0.0239  0.5452  3.9858
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.567160   0.504604  -1.124 0.261299
## fage            0.007768   0.006627   1.172 0.241437
## mage            0.003895   0.007050   0.552 0.580747
## weeks           0.191886   0.012176  15.759  < 2e-16 ***
## marital         0.097457   0.069384   1.405 0.160457
## gained          0.006913   0.002064   3.350 0.000839 ***
## lowbirthweight -2.289037   0.114258 -20.034  < 2e-16 ***
## gender         -0.347577   0.057730  -6.021 2.45e-09 ***
## habit          -0.265317   0.087539  -3.031 0.002502 **
## whitemom        0.233267   0.067981   3.431 0.000626 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9055 on 986 degrees of freedom
## Multiple R-squared:  0.6412, Adjusted R-squared:  0.638
## F-statistic: 195.8 on 9 and 986 DF,  p-value: < 2.2e-16
```

We eliminated the "visits" variable, and the standard error and R-squared did not change significantly after fitting the model.

```
# remove "mage" variable
transformed_df2 <- subset(transformed_df2, select = -c(mage))

stepwise_model <- lm(weight ~., data = transformed_df2)
cat("Removed: 'mage'", "\n")
```

```
## Removed: 'mage'
```

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = weight ~ ., data = transformed_df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4937 -0.6009 -0.0221  0.5503  4.0060
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.530331   0.500006  -1.061 0.289108
## fage            0.010234   0.004896   2.091 0.036828 *
## weeks           0.191527   0.012155  15.758  < 2e-16 ***
## marital         0.109685   0.065737   1.669 0.095524 .
## gained          0.006849   0.002060   3.325 0.000916 ***
## lowbirthweight -2.289458   0.114215 -20.045  < 2e-16 ***
## gender         -0.345653   0.057605  -6.000 2.76e-09 ***
## habit          -0.267501   0.087419  -3.060 0.002273 **
## whitemom        0.234891   0.067893   3.460 0.000564 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.9052 on 987 degrees of freedom
## Multiple R-squared:  0.6411, Adjusted R-squared:  0.6382
## F-statistic: 220.4 on 8 and 987 DF,  p-value: < 2.2e-16
```

We excluded the "mage" variable, and the standard error and R-squared showed no significant changes after fitting the model. However, it's worth noting that the p-value for "fage" has now dropped below 0.05.

```
# remove "marital" variable
transformed_df2 <- subset(transformed_df2, select = -c(marital))

stepwise_model <- lm(weight ~., data = transformed_df2)
cat("Removed: 'marital'", "\n")
```

```
## Removed: 'marital'
```

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = weight ~ ., data = transformed_df2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4611 -0.6123 -0.0276  0.5643  3.9245
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.542996   0.500399  -1.085  0.27813
## fage            0.012475   0.004712   2.648  0.00824 **
## weeks           0.191260   0.012164  15.723  < 2e-16 ***
## gained          0.006879   0.002062   3.337  0.00088 ***
## lowbirthweight -2.306724   0.113848 -20.261  < 2e-16 ***
## gender         -0.345822   0.057657  -5.998 2.80e-09 ***
## habit          -0.282968   0.087005  -3.252  0.00118 **
## whitemom        0.270478   0.064515   4.192 3.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.906 on 988 degrees of freedom
## Multiple R-squared:  0.6401, Adjusted R-squared:  0.6376
## F-statistic:   251 on 7 and 988 DF,  p-value: < 2.2e-16
```

We eliminated the "marital" variable, and the standard error and R-squared did not change significantly after fitting the model.

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = weight ~ ., data = transformed_df2)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -2.4611 -0.6123 -0.0276  0.5643  3.9245
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.542996   0.500399  -1.085  0.27813
## fage            0.012475   0.004712   2.648  0.00824 **
## weeks           0.191260   0.012164  15.723  < 2e-16 ***
## gained          0.006879   0.002062   3.337  0.00088 ***
## lowbirthweight -2.306724   0.113848 -20.261  < 2e-16 ***
## gender         -0.345822   0.057657  -5.998 2.80e-09 ***
## habit          -0.282968   0.087005  -3.252  0.00118 **
## whitemom        0.270478   0.064515   4.192 3.01e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.906 on 988 degrees of freedom
## Multiple R-squared:  0.6401, Adjusted R-squared:  0.6376
## F-statistic:   251 on 7 and 988 DF,  p-value: < 2.2e-16
```

Now, the final model reveals that the coefficients with a significant p-value < 0.05 in the multiple regression model are for the variables: "fage", "weeks," "gained," "lowbirthweight," "gender," "habit," and "whitemom.

We can present the model as the following formula:

weight = -0.54 + 0.01 * fage + 0.19 * weeks + 0.006 * gained - 2.30 * lowbirthweight - 0.35 * gender - 0.28 * habit + 0.27 * whitemom

Q5. Use the following data to predict the birth weight using the final model from question 4 above: fage = 40, mage = 32, mature = 'mature mom', weeks = 42, premie = 'full term', visits = 12, marital = 'married', gained=22, lowbirthweight = 'not low', gender = 'female', habit = 'nonsmoker', whitemom = 'white'. After which, derive the 95% confidence and prediction intervals for the forecasted birth weight. Comment on the results.

```r
# Create a data frame for prediction
new_data <- data.frame(
  fage = 40,
  mage = 32,
  mature = 1,
  weeks = 42,
  premie = 0,
  visits = 12,
  marital = 1,  #marital = 'married'
  gained = 22,
  lowbirthweight = 0, #lowbirthweight = 'not low'
  gender = 1,  #gender = 'female'
  habit = 0, #habit = 'nonsmoker'
```

```
  whitemom = 1 # whitemom = 'white'
)

# Predict birth weight
predicted_weight <- predict(stepwise_model, newdata = new_data)

confidence_interval <- predict(stepwise_model, newdata = new_data, interval = "confidence")

prediction_interval <- predict(stepwise_model, newdata = new_data, interval = "predict")

# Display the results
cat("Predicted weight:", predicted_weight, "\n")
```

```
## Predicted weight: 8.064913
```

```
cat("95% Confidence Interval:", confidence_interval[2], "to", confidence_interval[3], "\n")
```

```
## 95% Confidence Interval: 7.914663 to 8.215163
```

```
cat("95% Prediction Interval:", prediction_interval[2], "to", prediction_interval[3], "\n")
```

```
## 95% Prediction Interval: 6.280733 to 9.849093
```

The model predicts a birth weight of approximately 8.065 with a 95% confidence interval between 7.915 and 8.215. The prediction interval indicates that 95% of the birth weight will be within the range of 6.281 to 9.849. The narrower confidence interval reflects the precision of the model in estimating the mean birth weight, while the wider prediction interval acknowledges the additional variability in individual predictions. The difference between the two intervals emphasizes the uncertainty associated with predicting individual birth weights compared to estimating the mean.