Regression_Basics

```r
# Interested in predicting the amount of carbohydrates (in grams) for a menu item
# based on its calorie content (measured in 100s).

#install readxl package first
library(readxl)
coffee <- read_excel("starbucks.xlsx", na="NA", col_names = TRUE)
attach(coffee)

# Descriptive statistics
library(pastecs)
```

```
## Warning: package 'pastecs' was built under R version 3.3.3
```

```
## Loading required package: boot
```

```r
stat.desc(calories)
```

```
##      nbr.val     nbr.null       nbr.na          min          max
##   77.0000000    0.0000000    0.0000000    0.8000000    5.0000000
##        range          sum       median         mean      SE.mean
##    4.2000000  260.9000000    3.5000000    3.3883117    0.1200788
## CI.mean.0.95          var      std.dev     coef.var
##    0.2391576    1.1102563    1.0536870    0.3109770
```

```r
summary(calories)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.800   3.000   3.500   3.388   4.200   5.000
```

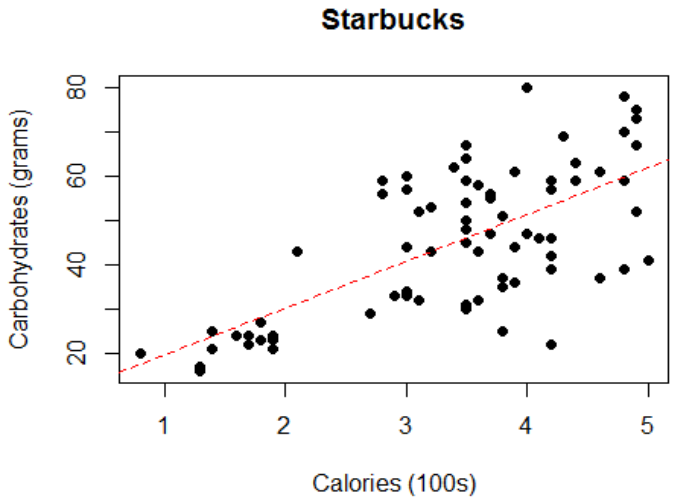
```r
stat.desc(carb)
```

```
##      nbr.val     nbr.null       nbr.na          min          max
##   77.0000000    0.0000000    0.0000000   16.0000000   80.0000000
##        range          sum       median         mean      SE.mean
##   64.0000000 3455.0000000   45.0000000   44.8701299    1.8862338
## CI.mean.0.95          var      std.dev     coef.var
##    3.7567602  273.9565960   16.5516342    0.3688787
```

```r
summary(carb)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00   31.00   45.00   44.87   59.00   80.00
```

```r
# A.
  # Scatter plot w/ fitted linear regression line
plot(calories, carb, pch = 16, main = "Starbucks", xlab = "Calories (100s)",
ylab = "Carbohydrates (grams)")
abline(lm(carb ~ calories), lty=2, col="red")
```




```r
# B/C/D/H
  # fit the model
    # linefit1 stores information that can be accessed
linefit1 <- lm(carb ~ calories)
    # to see an information summary of the fitted model
summary(linefit1)
```

```
##
## Call:
## lm(formula = carb ~ calories)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.477  -7.476  -1.029  10.127  28.644
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.944      4.746   1.884   0.0634 .
## calories      10.603      1.338   7.923 1.67e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.29 on 75 degrees of freedom
## Multiple R-squared:  0.4556, Adjusted R-squared:  0.4484
## F-statistic: 62.77 on 1 and 75 DF,  p-value: 1.673e-11
```

```r
      # check on p-value
2* (1 - pt(7.923, 75))
```

```
## [1] 1.671818e-11
```

```r
    # there are also functions for seeing specific features, e.g.,
    # to see the coefficients Beta-hats
coefficients(linefit1)
```

```
## (Intercept)    calories
##     8.94356    10.60309
```

```r
    # to see the coefficient of determination R-squared
summary(linefit1)$r.squared
```

```
## [1] 0.4556237
```

```r
    # standard deviation of the residuals = sqrt(MSE)
summary(linefit1)$sigma
```

```
## [1] 12.29325
```

```r
# E
# ANOVA data for simple linear regression
anova(linefit1)
```

```
## Analysis of Variance Table
##
## Response: carb
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## calories   1  9486.4  9486.4  62.772 1.673e-11 ***
## Residuals 75 11334.3   151.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# F.
  # Observed residuals, epsilon-hats
resids <- residuals(linefit1)
resids
```

```
##           1           2           3           4           5           6
##  20.9456291  17.9456291   5.5234670  14.1013049  -5.7275758  -1.1749887
##           7           8           9          10          11          12
##   3.2822315   6.8250113  -9.8131355 -14.4765330   1.7647024  10.1265557
##          13          14          15          16          17          18
. . .
##   2.5739686  -4.3559153 -20.7177685  -4.1146798  -6.2956064   1.9456291
##          73          74          75          76          77
##  -7.4765330 -14.2956064   3.2471734  16.2471734  19.2471734
```

```r
  # adding the residuals to the coffee data frame
coffee$resids <- resids
  # Use the data frame to identify specific residuals
subset(coffee, subset=item=="Apple Bran Muffin", select = c('resids'))
```

```
##     resids
## 2 17.94563
```

```r
subset(coffee, subset=item=="Apple Fritter", select = c('resids'))
```

```
##     resids
## 3 5.523467
```

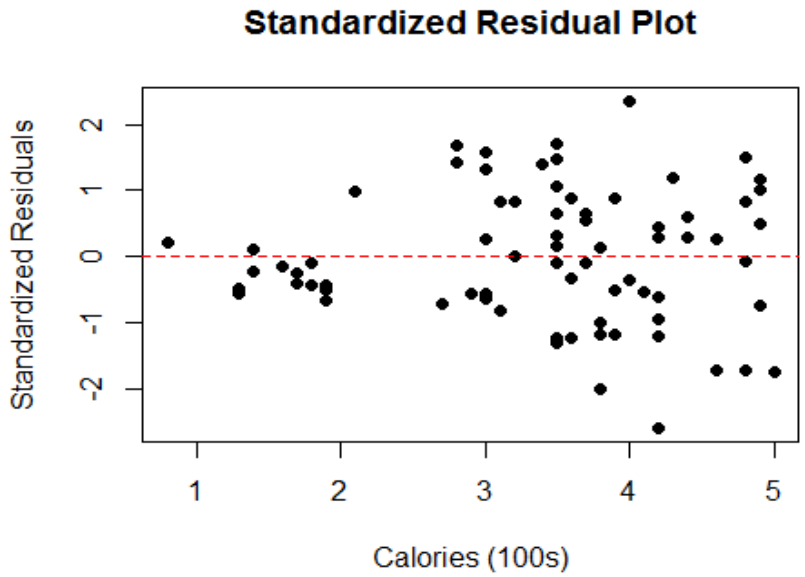
```r
# G.
  # Point estimte using fitted line
linefit1$coefficients[1] +linefit1$coefficients[2] * 2
```

```
## (Intercept)
##    30.14974
```

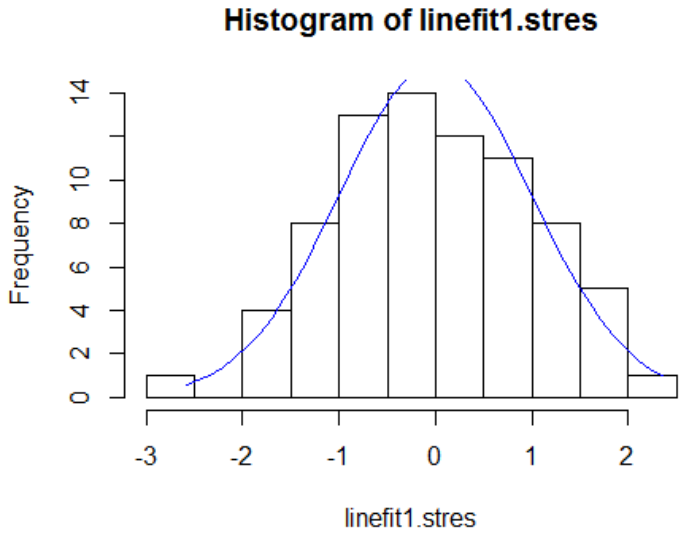
```r
# I.
  # confidence intervals for Beta-i
confint(linefit1, level = .90)
```

```
##                   5 %     95 %
## (Intercept) 1.039446 16.84767
## calories    8.374277 12.83190
```

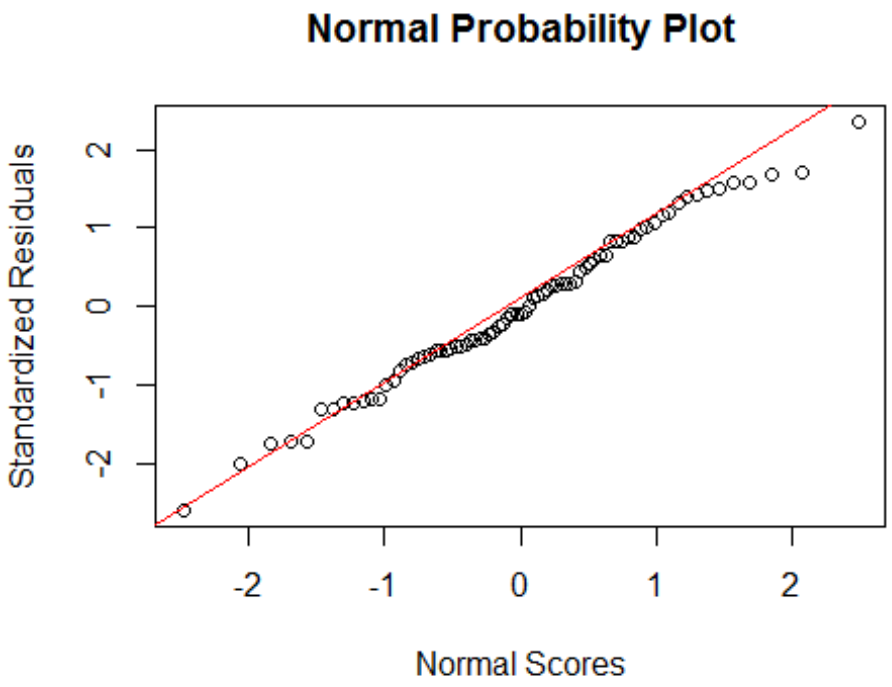
```r
# K. Assumption checking
  # standardized residual plot
linefit1.stres <- rstandard(linefit1)
plot(calories, linefit1.stres, pch = 16, main = "Standardized Residual Plot",
xlab = "Calories (100s)", ylab = "Standardized Residuals")
abline(0,0, lty=2, col="red")
```




```r
  # Normality
    # Histogram of standardized residuals with normal curve
h <- hist(linefit1.stres)
x <- linefit1.stres
xfit <- seq(min(x), max(x), length = 40)
yfit <- dnorm(xfit, mean = mean(x), sd = sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue")
```



```r
    # normal probability plot
qqnorm(linefit1.stres, main = "Normal Probability Plot", xlab = "Normal Score
s", ylab = "Standardized Residuals")
qqline(linefit1.stres, col = "red")
```




```r
    # hypothesis test
shapiro.test(linefit1.stres)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  linefit1.stres
## W = 0.99032, p-value = 0.832
```

```r
# L.
  # NOTE: Syntax below works if use attach followed by the lm syntax used abo
ve to create linefit1
    # create data frame with value of calories = 4.5 for which estimates/pred
ictions is desired
newdata <- data.frame(calories = 4.5)
    # 90% prediction interval for Y
predict(linefit1, newdata, interval="predict", level = .90)
```

```
##        fit      lwr     upr
## 1 56.65746 35.90302 77.4119
```

```r
    # for comparison: 90% confidence interval for the mean
predict(linefit1, newdata, interval="confidence", level = .90)
```

```
##        fit      lwr      upr
## 1 56.65746 53.25409 60.06083
```

```r
  # NOTE: As alternative (without attach) can use:
linefit2 <- lm(carb ~ calories, data = coffee)
predict(linefit2, newdata, interval="predict", level = .90)
```

```
##        fit      lwr     upr
## 1 56.65746 35.90302 77.4119
```

```r
# Clean up
detach(coffee)
```