Wine analytics Notebook

```r
rm(list=ls())
setwd("~/Documents/SUTD/Term 6/TAE/W2")
#install.packages("ggplot2")
if (!require(ggplot2)) {
  install.packages("ggplot2")
  library(ggplot2)
}
```

```
## Loading required package: ggplot2
```

```r
if (!require(psych)) {
  install.packages("psych")
  library(psych)
}
```

```
## Loading required package: psych

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
if (!require(ggfortify)) {
  install.packages("ggfortify")
  library(ggfortify)
}
```

```
## Loading required package: ggfortify
```

```r
if (!require(GGally)) {
  install.packages(("GGally"))
  library(GGally)
}
```

```
## Loading required package: GGally

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

The dataset consists of 38 observations of 6 variables (small dataset): 1952 to 1989. Ashenfelter published his paper in 1990.

```r
wine <- read.csv("wine.csv")
str(wine)
```

```
## 'data.frame':    38 obs. of  6 variables:
##  $ VINT   : int  1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 ...
##  $ LPRICE : num  -0.999 -0.454 NA -0.808 NA ...
##  $ WRAIN  : int  600 690 430 502 440 420 582 485 763 830 ...
##  $ DEGREES: num  17.1 16.7 15.4 17.1 15.7 ...
##  $ HRAIN  : int  160 80 180 130 140 110 187 187 290 38 ...
##  $ TIME_SV: int  31 30 29 28 27 26 25 24 23 22 ...
```

```r
summary(wine)
```

```
##       VINT          LPRICE           WRAIN          DEGREES
##  Min.   :1952   Min.   :-2.289   Min.   :376.0   Min.   :14.98
##  1st Qu.:1961   1st Qu.:-1.985   1st Qu.:510.2   1st Qu.:16.18
##  Median :1970   Median :-1.509   Median :586.5   Median :16.54
##  Mean   :1970   Mean   :-1.452   Mean   :605.0   Mean   :16.57
##  3rd Qu.:1980   3rd Qu.:-1.052   3rd Qu.:713.5   3rd Qu.:17.09
##  Max.   :1989   Max.   : 0.000   Max.   :845.0   Max.   :18.50
##                 NA's   :11
##      HRAIN          TIME_SV
##  Min.   : 38.0   Min.   :-6.00
##  1st Qu.: 87.5   1st Qu.: 3.25
##  Median :120.5   Median :12.50
##  Mean   :137.0   Mean   :12.50
##  3rd Qu.:171.0   3rd Qu.:21.75
##  Max.   :292.0   Max.   :31.00
##
```

```r
head(wine)
```

```
##   VINT   LPRICE WRAIN DEGREES HRAIN TIME_SV
## 1 1952 -0.99868   600 17.1167   160      31
## 2 1953 -0.45440   690 16.7333    80      30
## 3 1954       NA   430 15.3833   180      29
## 4 1955 -0.80796   502 17.1500   130      28
## 5 1956       NA   440 15.6500   140      27
## 6 1957 -1.50926   420 16.1333   110      26
```

1954 and 1956 wine prices are not available in the dataset since they are rarely sold now. The prices from 1981 to 1989 are not available in the dataset.

```r
is.na(wine)
```

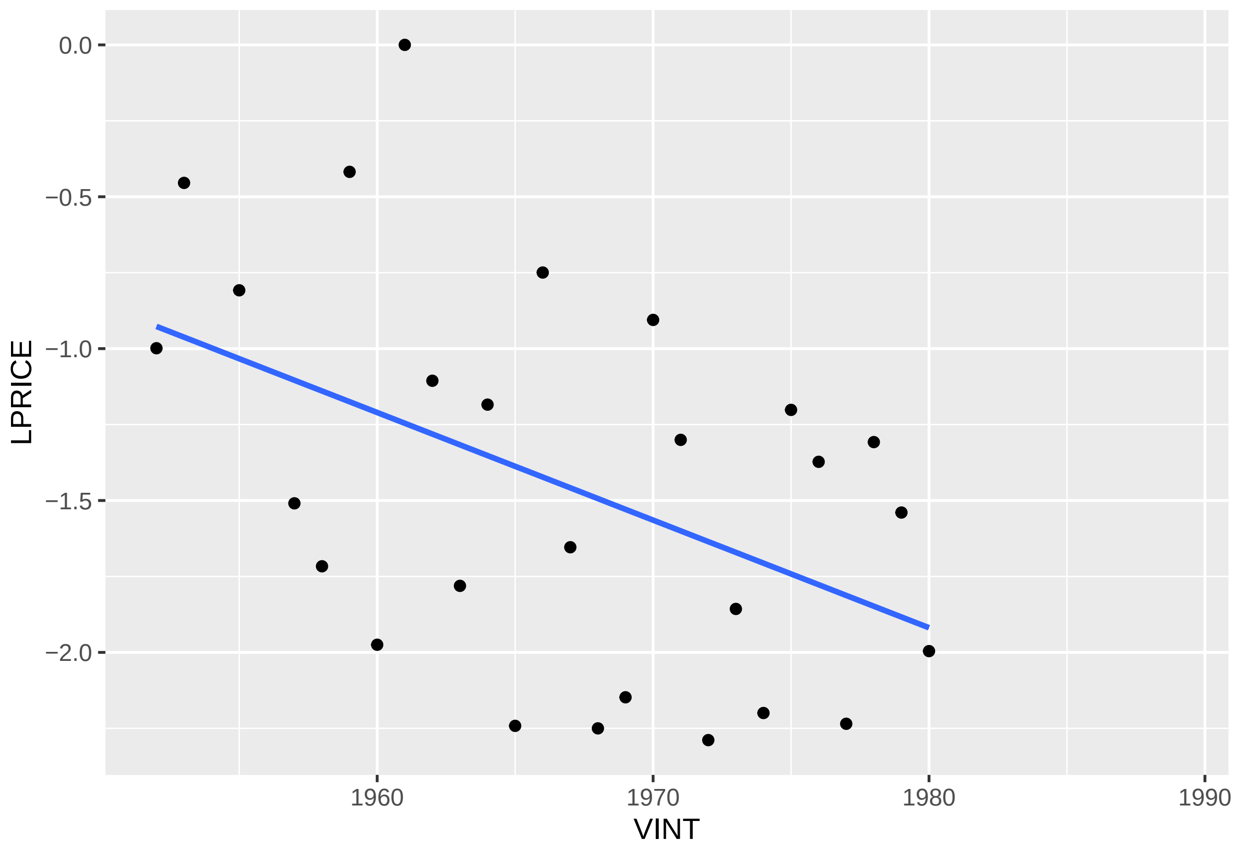
```
##        VINT LPRICE WRAIN DEGREES HRAIN TIME_SV
##  [1,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
##  [2,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
##  [3,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
##  [4,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
##  [5,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
```

```
##  [6,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
##  [7,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
##  [8,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
##  [9,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [10,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [11,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [12,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [13,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [14,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [15,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [16,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [17,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [18,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [19,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [20,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [21,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [22,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [23,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [24,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [25,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [26,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [27,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [28,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [29,] FALSE  FALSE FALSE   FALSE FALSE   FALSE
## [30,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [31,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [32,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [33,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [34,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [35,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [36,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [37,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
## [38,] FALSE   TRUE FALSE   FALSE FALSE   FALSE
```

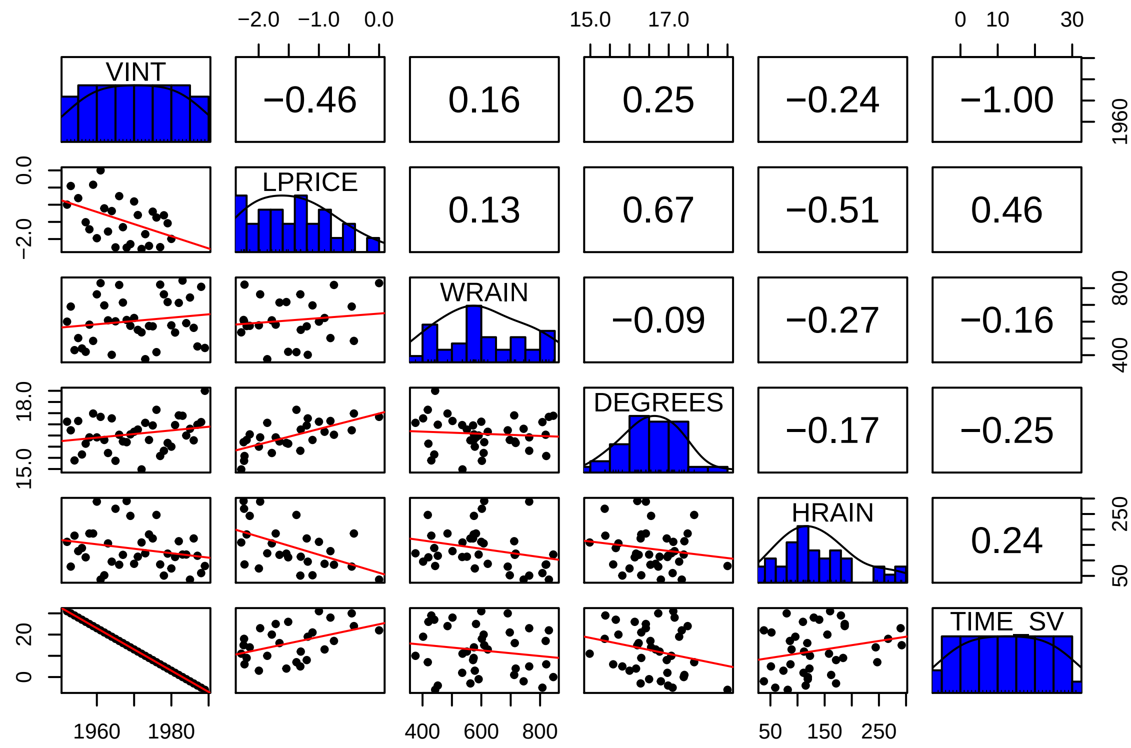
The scatter plot indicates a negative relationship but there is considerable variation that still needs to be captured. We can also plot a matrix of all scatter plots using the pairs command. Use plot() or ggplot() pairs.panels() or pairs()

```r
ggplot(wine, aes(x = VINT, y = LPRICE)) + geom_point(na.rm = TRUE) + geom_smooth(method = "lm", na.rm =
```

```
## `geom_smooth()` using formula 'y ~ x'
```



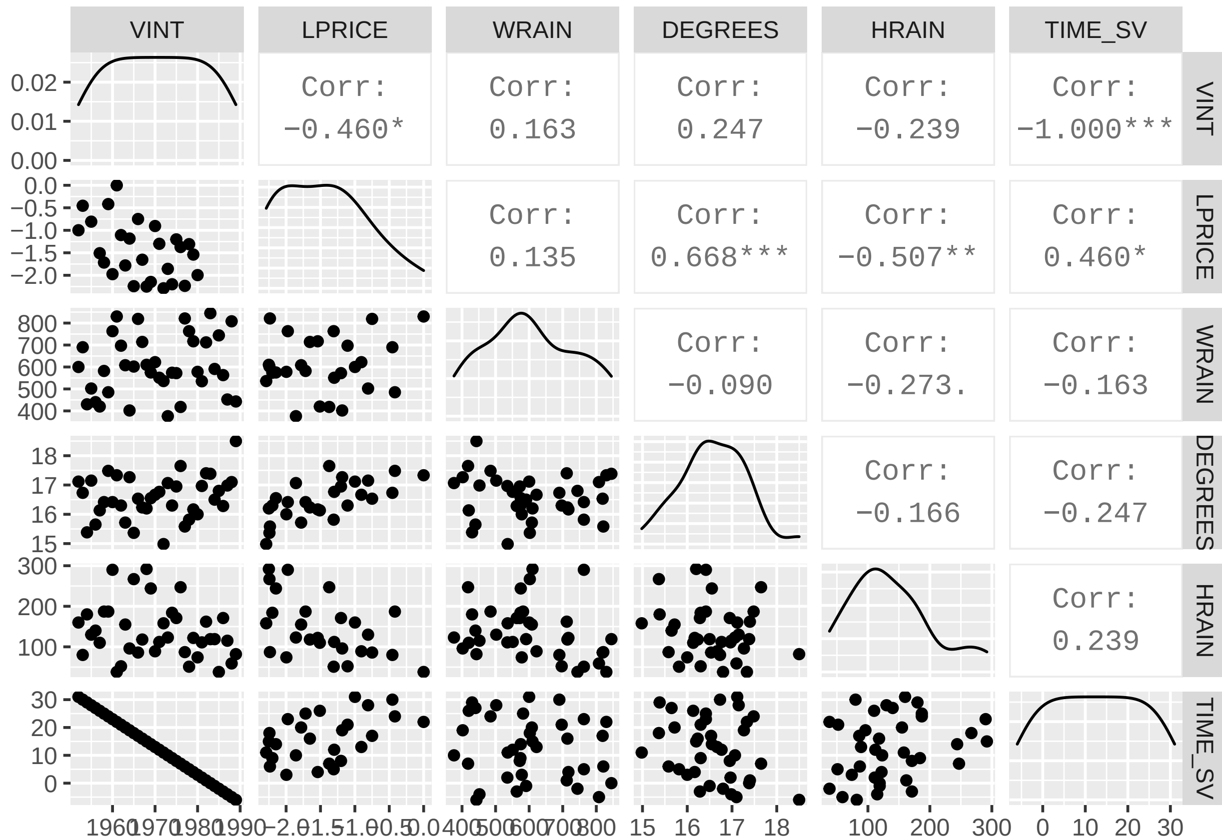

```r
# se = FALSE indicates not to display confidence interval

# to see the correlation between two variables
pairs.panels(wine, ellipses = F, lm = T, breaks = 10, hist.col = "blue") # uses Psych
```



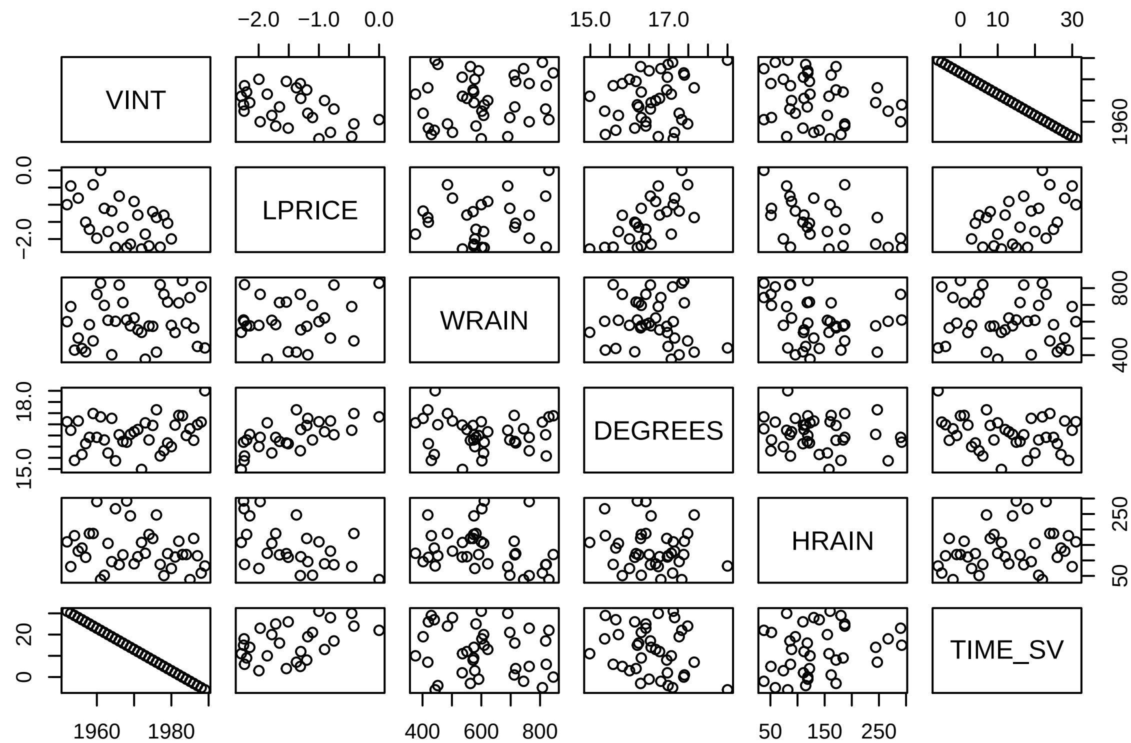

```r
ggpairs(wine) # uses GGally
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 11 rows containing missing values

## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 11 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 11 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 11 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 11 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 11 rows containing missing values

## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 11 rows containing missing values (geom_point).
```

```
## Warning: Removed 11 rows containing missing values (geom_point).

## Warning: Removed 11 rows containing missing values (geom_point).
```




```r
pairs(wine)
```



Split the data set into a training dataset from 1952 to 1978 (drop 1954 and 1956 since prices are not observable) and we use the test set from 1979 onwards. Note that for the test set however we only have prices till 1980 (so in this case we can only use 1979 and 1980) to test the model.

```
winetrain <- subset(wine, wine$VINT <= 1978 & !is.na(wine$LPRICE))
head(winetrain)
```

```
##   VINT   LPRICE WRAIN DEGREES HRAIN TIME_SV
## 1 1952 -0.99868   600 17.1167   160      31
## 2 1953 -0.45440   690 16.7333    80      30
## 4 1955 -0.80796   502 17.1500   130      28
## 6 1957 -1.50926   420 16.1333   110      26
## 7 1958 -1.71655   582 16.4167   187      25
## 8 1959 -0.41800   485 17.4833   187      24
```

```
winetest <- subset(wine, wine$VINT > 1978)
head(winetest)
```

```
##    VINT   LPRICE WRAIN DEGREES HRAIN TIME_SV
## 28 1979 -1.53960   717 16.1667   122       4
## 29 1980 -1.99582   578 16.0000    74       3
## 30 1981       NA   535 16.9667   111       2
## 31 1982       NA   712 17.4000   162       1
## 32 1983       NA   845 17.3833   119       0
## 33 1984       NA   591 16.5000   119      -1
```

One variable regression - lm() is the basic command to fit a linear model to the data. conclusion: this model might be missing something R-squared and adjusted values are not very indicative of a trend

```r
?lm # fitting linear model
# formula = Y ~ model
# in this case, Y ~ X
model1  <- lm(formula = LPRICE ~ VINT, data = winetrain)
model1
```

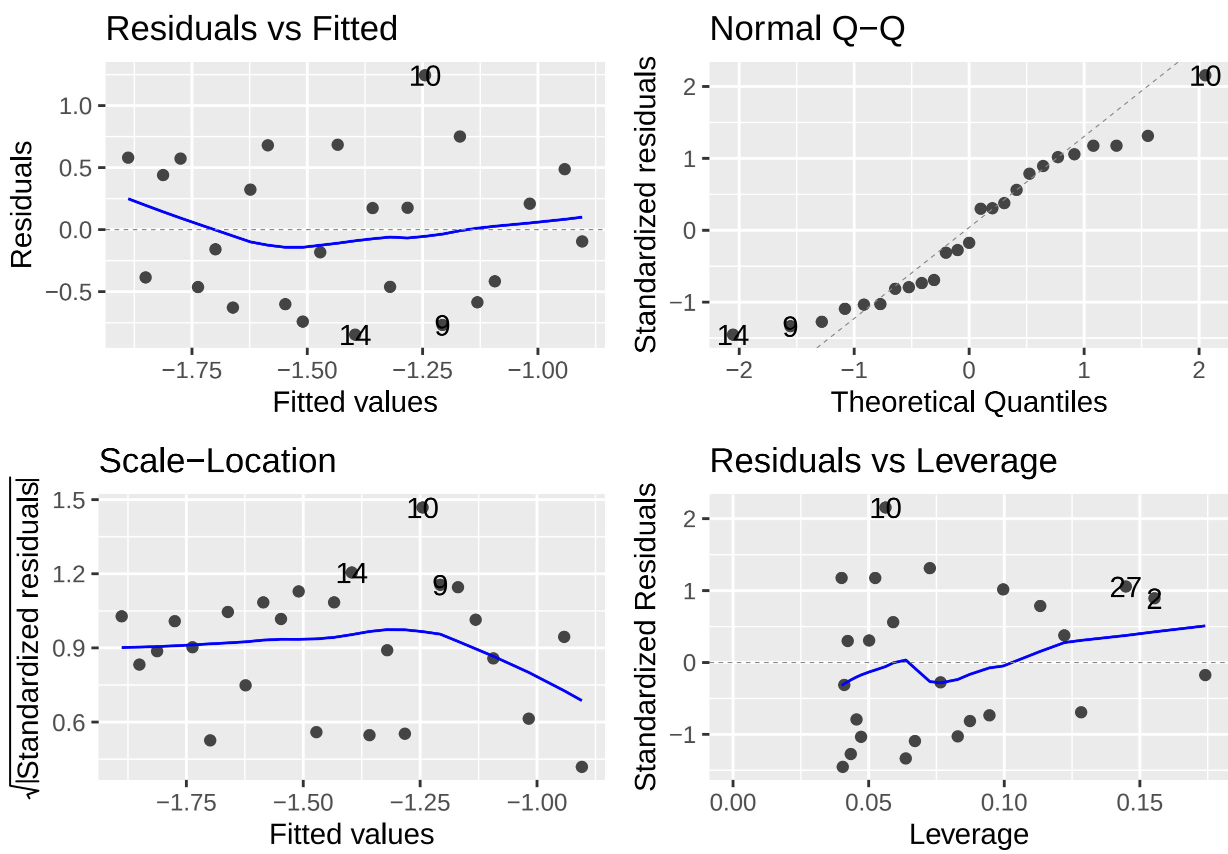
```
##
## Call:
## lm(formula = LPRICE ~ VINT, data = winetrain)
##
## Coefficients:
## (Intercept)         VINT
##    72.99301     -0.03786
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = LPRICE ~ VINT, data = winetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84574 -0.46266 -0.09462  0.48752  1.24478
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 72.99301   30.98789   2.356   0.0274 *
## VINT        -0.03786    0.01576  -2.402   0.0248 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.594 on 23 degrees of freedom
## Multiple R-squared:  0.2005, Adjusted R-squared:  0.1657
## F-statistic: 5.768 on 1 and 23 DF,  p-value: 0.0248
```

```r
autoplot(model1)
```

```
## Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
## Please use `arrange()` instead.
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```



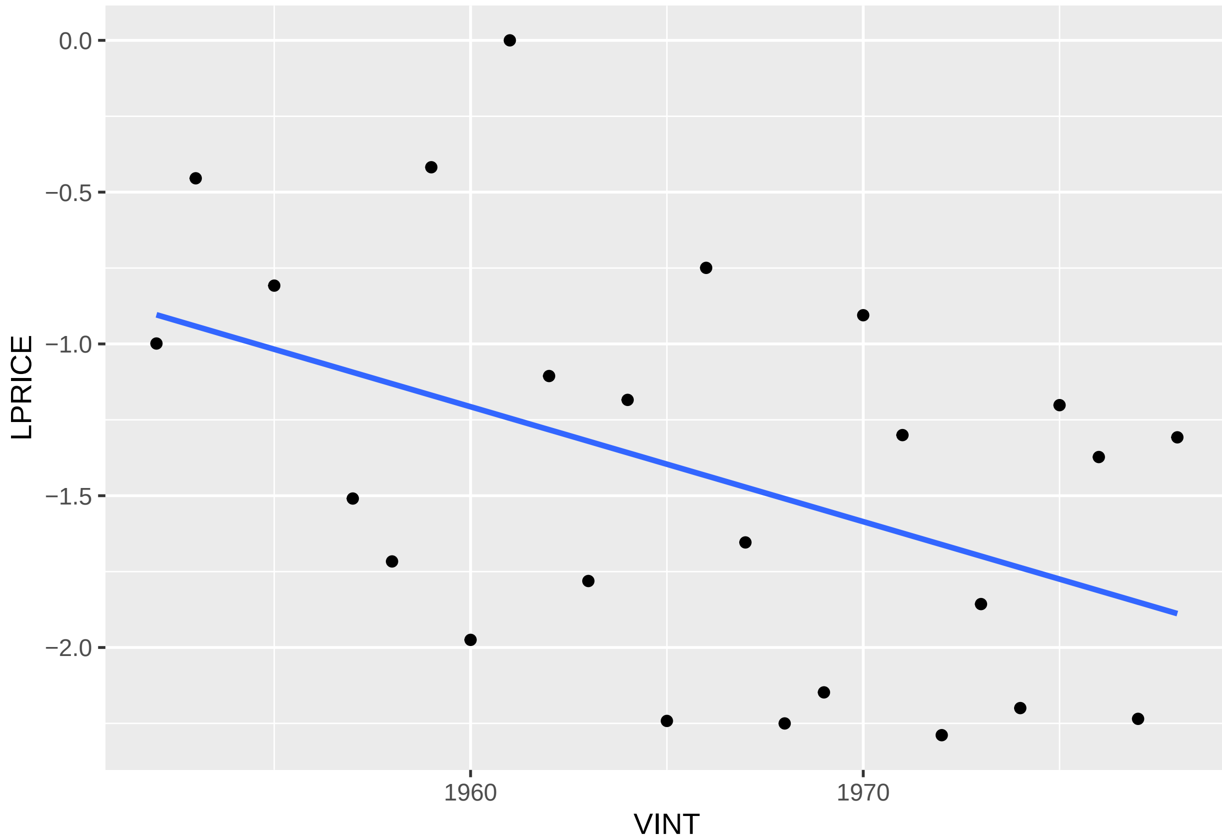
The regression fit here is LPRICE = 72.99 - 0.0378 * VINT. Both estimated coefficents are significant at the 0.01 level with R^2 = 0.2005 and adjusted R^2 = 0.1657. Plot the best fit line with a slope of -0.0378.

```
model1$coefficients
```

```
## (Intercept)        VINT
##  72.9930059  -0.0378571
```

```
ggplot(winetrain, aes(x = VINT, y = LPRICE)) + geom_point(na.rm = TRUE) + geom_smooth(method = "lm", na
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Evaluate the sum of squared errors and total sum of squares

residuals = actual - fitted

```
sse1 <- sum(model1$residuals^2)
sse1
```

```
## [1] 8.115495
```

```
sst1 <- sum((winetrain$LPRICE - mean(winetrain$LPRICE))^2)
sst1
```

```
## [1] 10.15058
```

```
1 - sse1/sst1 # 0.2004897
```

```
## [1] 0.2004897
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = LPRICE ~ VINT, data = winetrain)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84574 -0.46266 -0.09462  0.48752  1.24478
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 72.99301   30.98789   2.356   0.0274 *
## VINT        -0.03786    0.01576  -2.402   0.0248 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.594 on 23 degrees of freedom
## Multiple R-squared:  0.2005, Adjusted R-squared:  0.1657
## F-statistic: 5.768 on 1 and 23 DF,  p-value: 0.0248
```

The result indicates that older the wine, greater is the value but there is still significant variation.

One variable regression - continued

- X: WRAIN
- intercept is of significance but the variable `WRAIN` is not significant
- p-value is too high and low R squared value.

```
model2  <- lm(LPRICE ~ WRAIN, data = winetrain)
summary(model2)
```

```
##
## Call:
## lm(formula = LPRICE ~ WRAIN, data = winetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95348 -0.65439  0.04172  0.43136  1.27550
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.8331730  0.6286160  -2.916  0.00777 **
## WRAIN        0.0006719  0.0010155   0.662  0.51479
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6581 on 23 degrees of freedom
## Multiple R-squared:  0.01868,    Adjusted R-squared:  -0.02399
## F-statistic: 0.4377 on 1 and 23 DF,  p-value: 0.5148
```

```
model3  <- lm(LPRICE ~ HRAIN, data = winetrain)
summary(model3)
```

```
##
## Call:
## lm(formula = LPRICE ~ HRAIN, data = winetrain)
```

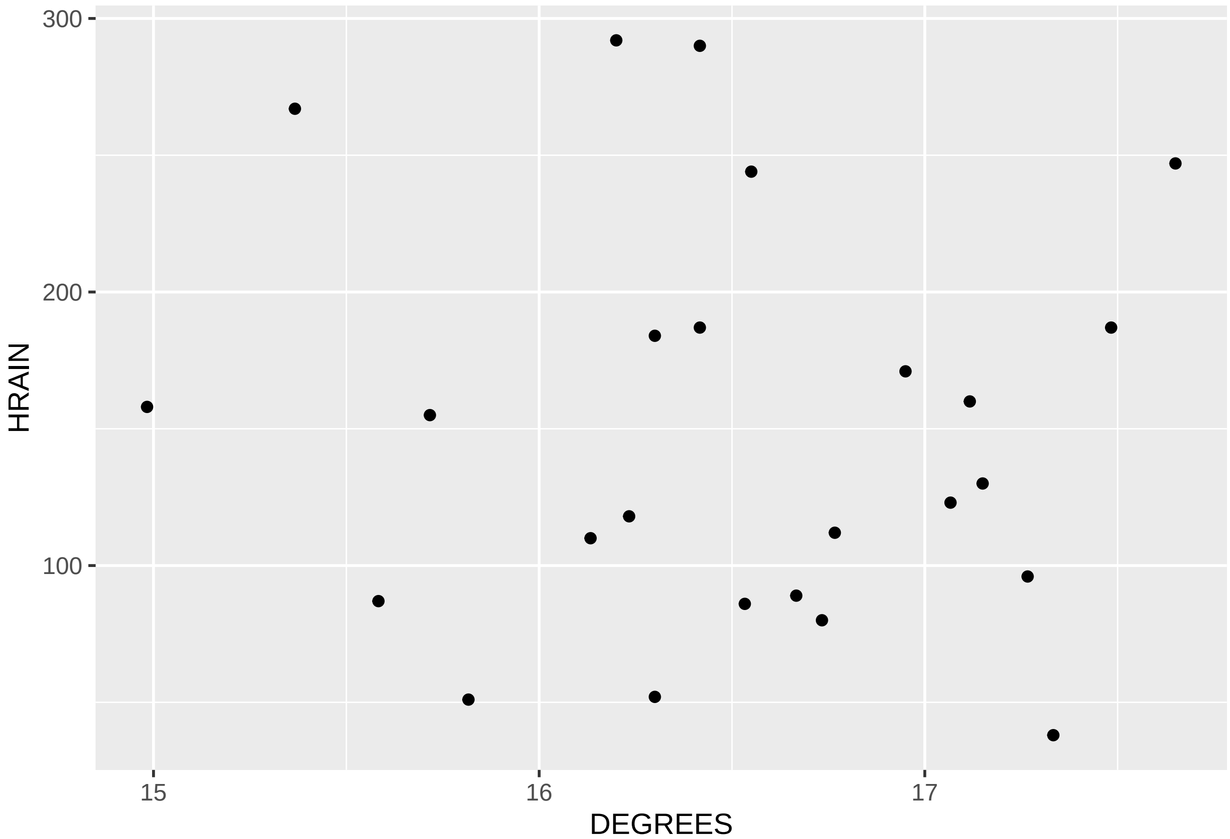
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1116 -0.3228 -0.1008  0.3691  1.1977
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.695162   0.249152   -2.79  0.01040 *
## HRAIN       -0.004923   0.001506   -3.27  0.00337 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5489 on 23 degrees of freedom
## Multiple R-squared:  0.3173, Adjusted R-squared:  0.2877
## F-statistic: 10.69 on 1 and 23 DF,  p-value: 0.003366
```

```r
model4  <- lm(LPRICE ~ DEGREES, data = winetrain)
summary(model4)
```

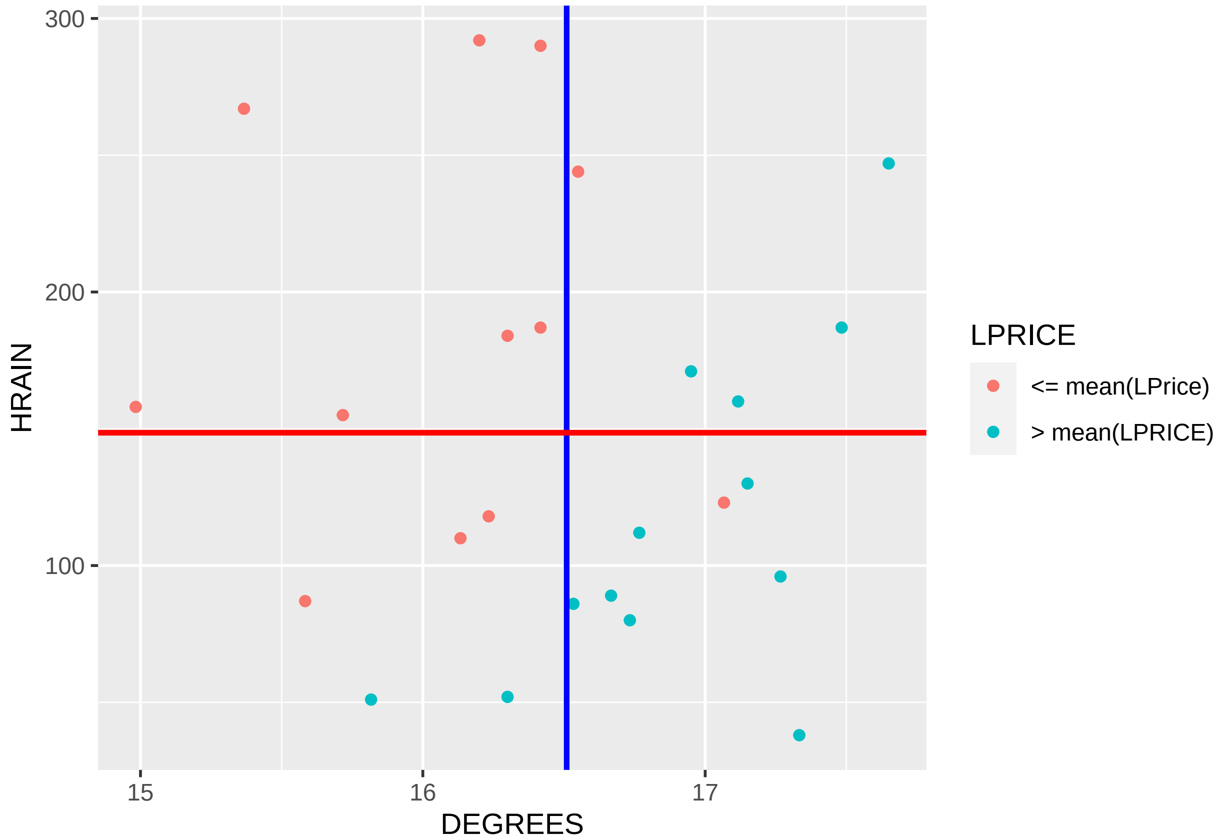
```
##
## Call:
## lm(formula = LPRICE ~ DEGREES, data = winetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78449 -0.23885 -0.03727  0.38994  0.90320
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.9114     2.4935  -4.777 8.12e-05 ***
## DEGREES       0.6351     0.1509   4.208 0.000335 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4993 on 23 degrees of freedom
## Multiple R-squared:  0.435,  Adjusted R-squared:  0.4105
## F-statistic: 17.71 on 1 and 23 DF,  p-value: 0.0003351
```

Two variables - The effect of DEGREES and HRAIN on LPRICE.

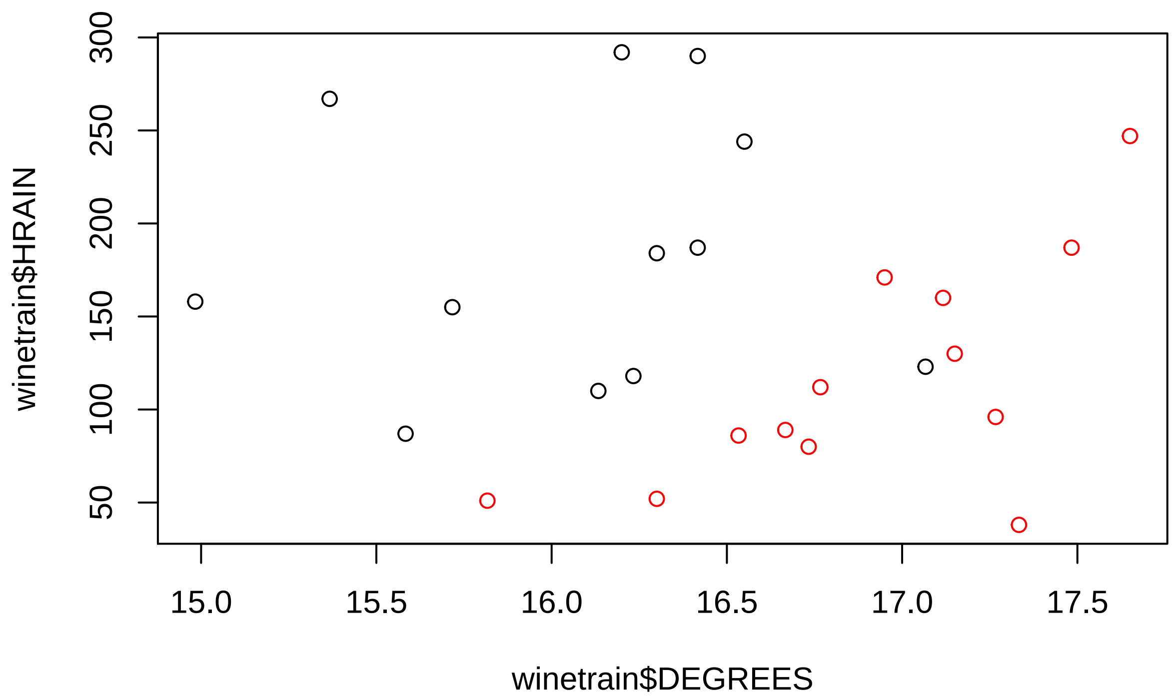
```r
ggplot(winetrain, aes(x = DEGREES, y = HRAIN)) + geom_point(na.rm = TRUE)
```




```
br <- mean(winetrain$LPRICE)

ggplot(winetrain, aes(x = DEGREES, y = HRAIN, color = cut(LPRICE, c(-Inf, -1.42, Inf)))) + geom_point(na
```




```
plot(x = winetrain$DEGREES, y = winetrain$HRAIN, col = ifelse(winetrain$LPRICE >= mean(winetrain$LPRICE)
```



The figure indicates that hot and dry summers produce wines that obtain higher prices while cooler summers with more rain gives lower priced wines. 1961 is an year where an extremely high quality wine was produced.

Two variable regression

```r
model5  <- lm(LPRICE ~ DEGREES + HRAIN, data = winetrain)
summary(model5)
```

```
##
## Call:
## lm(formula = LPRICE ~ DEGREES + HRAIN, data = winetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88319 -0.19599  0.06181  0.15379  0.59724
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.69628    1.85444  -5.768 8.40e-06 ***
## DEGREES       0.60261    0.11128   5.415 1.94e-05 ***
## HRAIN        -0.00457    0.00101  -4.525 0.000167 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3674 on 22 degrees of freedom
## Multiple R-squared:  0.7074, Adjusted R-squared:  0.6808
```

```
## F-statistic: 26.59 on 2 and 22 DF,  p-value: 1.348e-06
```

LPRICE = -10.69 + 0.602*DEGREES - 0.0045*HRAIN. Both variables are extremely significant in the fit with R^2 = 0.7 and adjusted R^2 = 0.68.

Multiple linear regression

```r
# use all variables as X input
model6  <- lm(LPRICE ~ ., data = winetrain)
summary(model6)
```

```
##
## Call:
## lm(formula = LPRICE ~ ., data = winetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45473 -0.24276  0.00753  0.19770  0.53640
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.5297567 16.4435924   2.161 0.043028 *
## VINT        -0.0239302  0.0080969  -2.955 0.007821 **
## WRAIN        0.0010756  0.0005073   2.120 0.046684 *
## DEGREES      0.6072099  0.0987030   6.152  5.2e-06 ***
## HRAIN       -0.0039715  0.0008538  -4.652 0.000154 ***
## TIME_SV             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.295 on 20 degrees of freedom
## Multiple R-squared:  0.8286, Adjusted R-squared:  0.7943
## F-statistic: 24.17 on 4 and 20 DF,  p-value: 2.036e-07
```

```r
cor(winetrain)
```

```
##                VINT     LPRICE       WRAIN     DEGREES       HRAIN     TIME_SV
## VINT     1.00000000 -0.4477608  0.01697002 -0.24691585  0.02800907 -1.00000000
## LPRICE  -0.44776081  1.0000000  0.13666199  0.65955892 -0.56332212  0.44776081
## WRAIN    0.01697002  0.1366620  1.00000000 -0.32109061 -0.27544085 -0.01697002
## DEGREES -0.24691585  0.6595589 -0.32109061  1.00000000 -0.06449593  0.24691585
## HRAIN    0.02800907 -0.5633221 -0.27544085 -0.06449593  1.00000000 -0.02800907
## TIME_SV -1.00000000  0.4477608 -0.01697002  0.24691585 -0.02800907  1.00000000
```

Note that TIME_SV coefficients are not defined as it is perfectly correlated with the VINT variable (perfect multicollinearity). We drop the variable and redo the regression. High correlation (in absolute value) between independent variables is not good (indication of multicollinearity) while high correlation (in absolute value) between dependent and independent variables is good.

```r
# since time_sv and VINT is perfectly collinear, we can just drop that variable in our model
# the same values are obtained as from above, beause the model automatically drops the perfectly collin
model7  <- lm(LPRICE ~ VINT + WRAIN + HRAIN + DEGREES, data = winetrain)
summary(model7)
```

```
##
## Call:
## lm(formula = LPRICE ~ VINT + WRAIN + HRAIN + DEGREES, data = winetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45473 -0.24276  0.00753  0.19770  0.53640
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.5297567 16.4435924   2.161 0.043028 *
## VINT        -0.0239302  0.0080969  -2.955 0.007821 **
## WRAIN        0.0010756  0.0005073   2.120 0.046684 *
## HRAIN       -0.0039715  0.0008538  -4.652 0.000154 ***
## DEGREES      0.6072099  0.0987030   6.152  5.2e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.295 on 20 degrees of freedom
## Multiple R-squared:  0.8286, Adjusted R-squared:  0.7943
## F-statistic: 24.17 on 4 and 20 DF,  p-value: 2.036e-07
```

R^2 = 0.828 and adjusted R^2 = 0.794. The coefficients indicate that high quality wines correlate strongly in a positive manner with summer temperatures, negatively correlate with harvest rain and positively correlate with winter rain. The result indicates that 80% of the variaion can be explained by including the weather variables in comparison to 20% with only the vintage year.

```r
model7a <- lm(LPRICE ~ WRAIN + DEGREES + HRAIN, data = winetrain)
summary(model7a)
```

```
##
## Call:
## lm(formula = LPRICE ~ WRAIN + DEGREES + HRAIN, data = winetrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67475 -0.12957  0.01975  0.20754  0.63848
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.280e+01  2.037e+00  -6.282 3.13e-06 ***
## WRAIN        1.177e-03  5.920e-04   1.987 0.060085 .
## DEGREES      6.810e-01  1.117e-01   6.097 4.75e-06 ***
## HRAIN       -3.948e-03  9.987e-04  -3.953 0.000726 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.345 on 21 degrees of freedom
## Multiple R-squared:  0.7537, Adjusted R-squared:  0.7185
## F-statistic: 21.42 on 3 and 21 DF,  p-value: 1.359e-06
```

We can run the model by dropping VINT but this decreases R^2 to 0.75 and adjusted R^2 to 0.71.

We can obtain confidence intervals for the estimates using the confint command.

```r
model7$coefficients
```

```
## (Intercept)         VINT        WRAIN        HRAIN      DEGREES
## 35.529756684 -0.023930151  0.001075566 -0.003971495  0.607209852
```

```r
model7$residuals
```

```
##            1            2            4            6            7            8
## -0.220111003  0.166382893  0.008439992 -0.018882929 -0.242762656  0.536397370
##            9           10           11           12           13           14
## -0.238875854  0.130516175 -0.125192530  0.082452451 -0.250914254  0.333136997
##           15           16           17           18           19           20
##  0.188967781 -0.269366853 -0.018617681 -0.257857910  0.271477579  0.007526308
##           21           22           23           24           25           26
##  0.324696753 -0.451555852 -0.275358804  0.302306645  0.197700658 -0.454730835
##           27
##  0.274225558
```

```r
# compute confidence interval around the beta value
confint(model7)
```

```
##                     2.5 %       97.5 %
## (Intercept)  1.229024e+00 69.830489379
## VINT        -4.082008e-02 -0.007040227
## WRAIN        1.739315e-05  0.002133739
## HRAIN       -5.752501e-03 -0.002190488
## DEGREES      4.013189e-01  0.813100762
```

```r
confint(model7, level = 0.99)
```

```
##                     0.5 %        99.5 %
## (Intercept) -1.125785e+01 82.3173631200
## VINT        -4.696870e-02 -0.0008916014
## WRAIN       -3.678252e-04  0.0025189572
## HRAIN       -6.400860e-03 -0.0015421290
## DEGREES      3.263662e-01  0.8880534986
```

Predictions - The predict function helps predict the outcome of the model on values in the test set. The test R-squared for model 7 is 0.82, for model 4 it is 0.435 and for model 5 it is 0.70.

```r
str(winetest)
```

```
## 'data.frame':    11 obs. of  6 variables:
##  $ VINT   : int  1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 ...
##  $ LPRICE : num  -1.54 -2 NA NA NA ...
##  $ WRAIN  : int  717 578 535 712 845 591 744 563 452 808 ...
##  $ DEGREES: num  16.2 16 17 17.4 17.4 ...
##  $ HRAIN  : int  122 74 111 162 119 119 38 171 115 59 ...
##  $ TIME_SV: int  4 3 2 1 0 -1 -2 -3 -4 -5 ...
```

```r
?predict
# newdata = test set
wineprediction7 <- predict(model7, newdata = winetest)
wineprediction7
```

```
##         28         29         30         31         32         33         34
## -1.7247744 -1.8087984 -1.4389334 -1.2119306 -0.9321766 -1.7656490 -1.1211635
##         35         36         37         38
## -2.1817252 -1.6775926 -1.0253562 -0.6831185
```

```r
# Year 1979 and 1980: winetest true LPRICE values, predicted values
cbind(c(1979, 1980), winetest$LPRICE[1:2], wineprediction7[1:2])
```

```
##    [,1]     [,2]      [,3]
## 28 1979 -1.53960 -1.724774
## 29 1980 -1.99582 -1.808798
```

```r
sse7 <- sum((wineprediction7[1:2] - winetest$LPRICE[1:2])^2)

# since i dont have a huge test set, take the mean of the training data set for calculation purpose
sst <- sum((winetest$LPRICE[1:2] - mean(winetrain$LPRICE))^2)

1 - sse7 / sst
```

```
## [1] 0.7944189
```

```r
wineprediction4 <- predict(model4, newdata = winetest)
sse4 <- sum((wineprediction4[1:2] - winetest$LPRICE[1:2])^2)
1 - sse4 / sst
```

```
## [1] 0.7881924
```

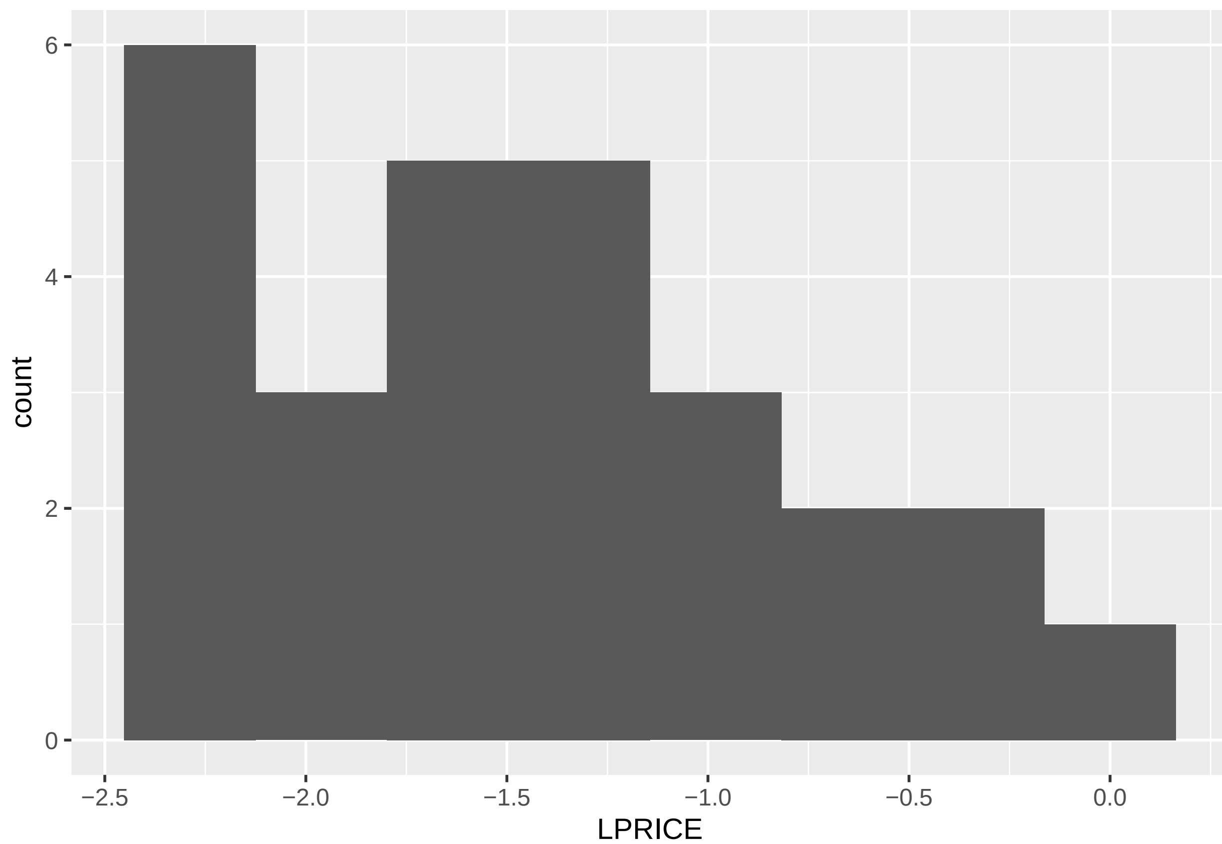
```r
wineprediction5 <- predict(model5,newdata=winetest)
sse5 <- sum((wineprediction5[1:2] - winetest$LPRICE[1:2])^2)
1 - sse5 / sst
```

```
## [1] -0.08201462
```

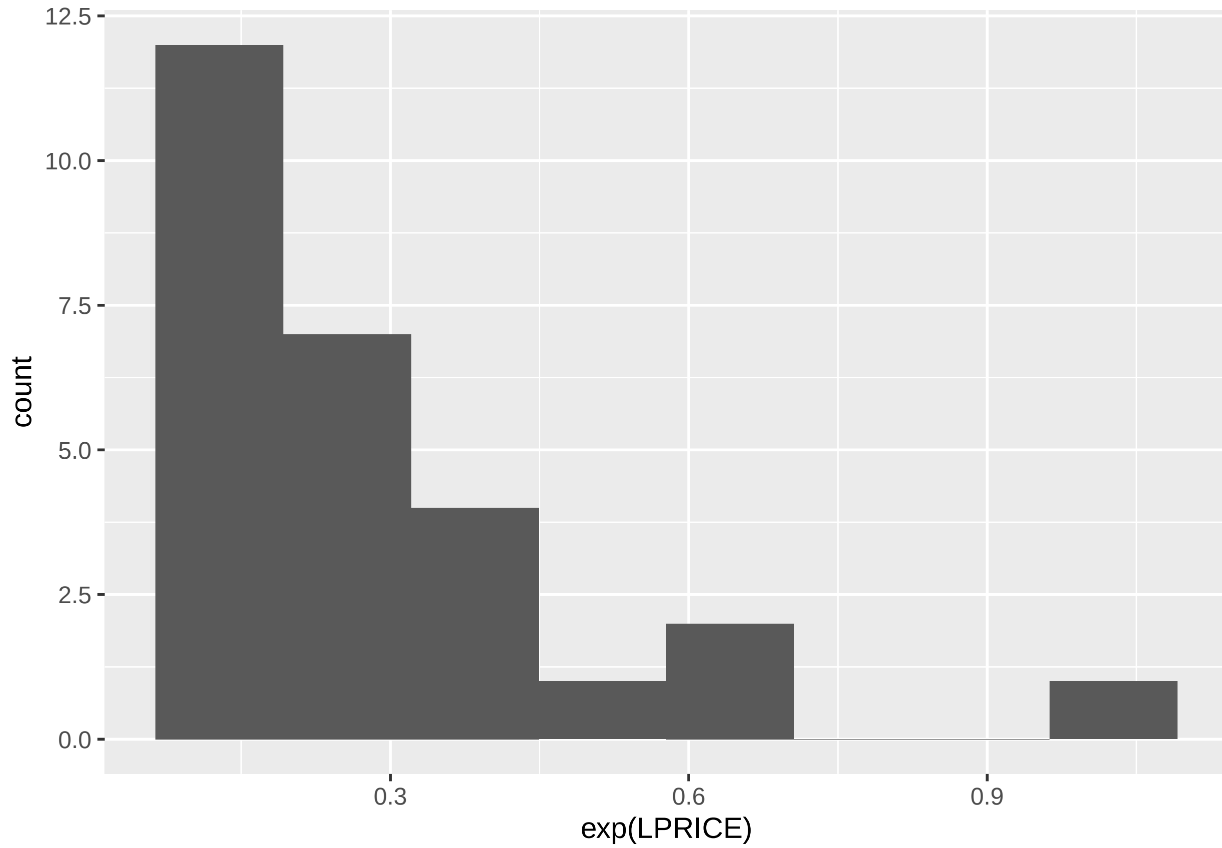
```r
# model5 might not be good because test set R^2 value is not good, although it is decent in train set
```

We use the training test mean to compute the total sum of squares for computing the test R^2 values. The results indicate that better R^2 in the training set does not necessarily indicate better R^2 in test set (this can also be negative).

```r
ggplot(wine, aes(x = LPRICE)) + geom_histogram(bins = 8, na.rm = TRUE)
```




```
ggplot(wine ,aes(x = exp(LPRICE))) + geom_histogram(bins = 8, na.rm = TRUE)
```




```
#hist(wine$LPRICE)
#hist(exp(wine$LPRICE))

# hist(wine$LPRICE)
# hist(exp(wine$LPRICE))
```

The use of logarithms for prices is fairly common in the economics literature - partly jusified by skewed values in some datasets dealing with numbers such as salaries and partly justified by functional relations such as y = exp(a+bx) which gives log y = a+bx. Such transformations need to be justified and is sometime specific to domains. Even if one directly uses prices in the regression, similar insights are found in this dataset.