Lab 10

Jennifer Lin

jenniferyjlin@berkeley.edu (mailto:jenniferyjlin@berkeley.edu)

```
library(datasets)
library(ggplot2)
```

1

Load the dataset iris

(a) Perform a linear regression where you aim to predict the pedal width from its pedal length

(b) Based off of your results, what would be the predicted pedal width of the iris if its pedal length is 2.3?

(c) Plot a histogram of the residuals of this analysis and perform a Shapiro-Wilk normality test

```
data(iris)
# (a)
vec_PL <- iris$Petal.Length
vec_PW <- iris$Petal.Width
result_iris <- lm(vec_PW ~ vec_PL)
print(result_iris)
```

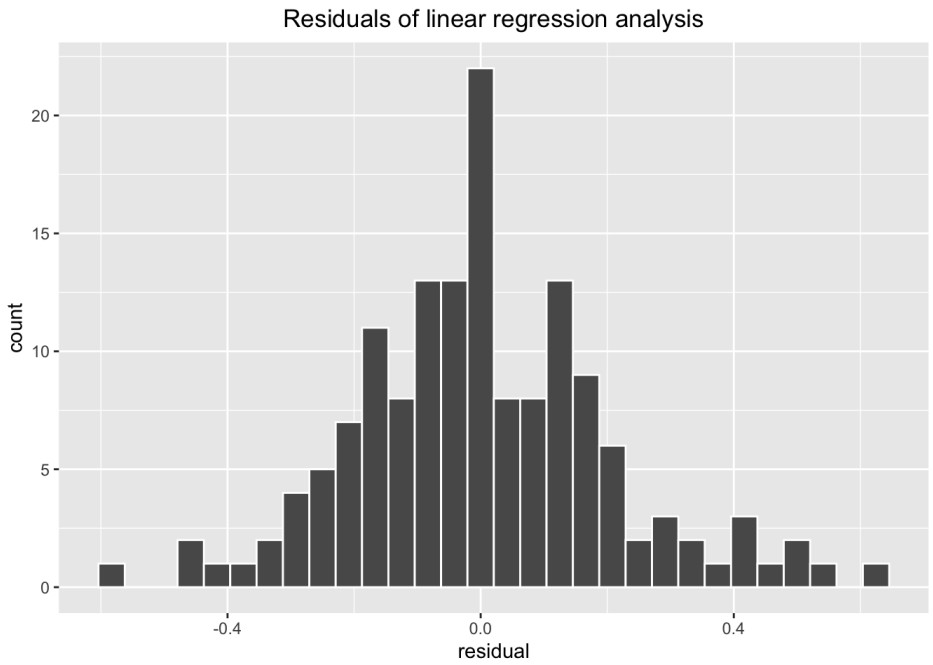
```
##
## Call:
## lm(formula = vec_PW ~ vec_PL)
##
## Coefficients:
## (Intercept)       vec_PL
##     -0.3631       0.4158
```

```
# (b)
intercept <- as.numeric(result_iris$coefficients[1])
slope <- as.numeric(result_iris$coefficients[2])
vec_PW_predict <- slope*2.3 + slope
print(vec_PW_predict)
```

```
## [1] 1.371993
```

```
# (c)
lm_residuals <- result_iris$residuals
p_lm_residuals <- ggplot() + aes(lm_residuals) + geom_histogram(colour="white") +
  labs(title="Residuals of linear regression analysis") +
  xlab("residual") +
  theme(plot.title = element_text(hjust = 0.5))
print(p_lm_residuals)
```

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




```
lm_residuals_St <- shapiro.test(lm_residuals)
print(lm_residuals_St)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lm_residuals
## W = 0.98378, p-value = 0.07504
```

```
print(lm_residuals_St$p.value)
```

```
## [1] 0.07503625
```

```
if(lm_residuals_St$p.value>0.05){
  print(paste0("The p value is ", round(lm_residuals_St$p.value,4),  ", which is above 0.05. Thus, our variable i
s normally distributed."))
}else{
  print(paste0("The p value is ", round(lm_residuals_St$p.value,4),  ", which is under 0.05. Thus, our variable i
s NOT normally distributed."))
}
```

```
## [1] "The p value is 0.075, which is above 0.05. Thus, our variable is normally distributed."
```

2

Load the dataset trees

(a) Perform a linear regression where you aim to predict the volume from both its girth and height

(b) Try seeing if it is more prudent use only one parameter (single variable regression) instead of both girth and height, and prove it by using the appropriate test

(c) Find the 95% confidence intervals for the regression coefficients. You may want to use bootstrap- ping to derive these values.

```
rm(list=ls())
data(trees)
# (a)
vec_G <- trees$Girth
vec_H <- trees$Height
vec_V <- trees$Volume
result_trees_V_GH <- lm(vec_V ~ vec_G+vec_H)
print(result_trees_V_GH)
```

```
##
## Call:
## lm(formula = vec_V ~ vec_G + vec_H)
##
## Coefficients:
## (Intercept)        vec_G        vec_H
##    -57.9877       4.7082       0.3393
```

```
# (b)
result_trees_V_G <- lm(vec_V ~ vec_G)
result_trees_V_H <- lm(vec_V ~ vec_H)
# Anova model comparison can test whether the more complex model is significantly better at capturing the data th
an the simpler model.
anova_result_GH_G <- anova(result_trees_V_GH,result_trees_V_G)
print(anova_result_GH_G)
```

```
## Analysis of Variance Table
##
## Model 1: vec_V ~ vec_G + vec_H
## Model 2: vec_V ~ vec_G
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     28 421.92
## 2     29 524.30 -1   -102.38 6.7943 0.01449 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
if(anova_result_GH_G$`Pr(>F)`[2]>0.05){
  print(paste0("The p value is ", round(anova_result_GH_G$`Pr(>F)`[2],4),  ", which is above 0.05. Thus, the more
complex model is NOT significantly better at capturing the data than the simpler model."))
}else{
  print(paste0("The p value is ", round(anova_result_GH_G$`Pr(>F)`[2],4),  ", which is under 0.05. Thus, the more
complex model is significantly better at capturing the data than the simpler model."))
}
```

```
## [1] "The p value is 0.0145, which is under 0.05. Thus, the more complex model is significantly better at captu
ring the data than the simpler model."
```

```
anova_result_GH_H <- anova(result_trees_V_GH,result_trees_V_H)
print(anova_result_GH_H)
```

```
## Analysis of Variance Table
##
## Model 1: vec_V ~ vec_G + vec_H
## Model 2: vec_V ~ vec_H
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     28  421.9
## 2     29 5204.9 -1     -4783 317.41 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
if(anova_result_GH_H$`Pr(>F)`[2]>0.05){
  print(paste0("The p value is ", round(anova_result_GH_H$`Pr(>F)`[2],4),  ", which is above 0.05. Thus, the more
complex model is NOT significantly better at capturing the data than the simpler model."))
}else{
  print(paste0("The p value is ", round(anova_result_GH_H$`Pr(>F)`[2],4),  ", which is under 0.05. Thus, the more
complex model is significantly better at capturing the data than the simpler model."))
}
```

```
## [1] "The p value is 0, which is under 0.05. Thus, the more complex model is significantly better at capturing
the data than the simpler model."
```

```
# AIC is a relative measure of model fit. It is used for model selection.
## Lower value of AIC suggests a better model.
AIC_GH <- AIC(result_trees_V_GH)
print(AIC_GH) # Better model (predict the volume from both its girth and height)
```

```
## [1] 176.91
```

```
AIC_G <- AIC(result_trees_V_G)
print(AIC_G)
```

```
## [1] 181.6447
```

```
AIC_H <- AIC(result_trees_V_H)
print(AIC_H)
```

```
## [1] 252.7986
```

```
if(min(AIC_GH,AIC_G,AIC_H)==AIC_GH){
  print("Predicting the volume from both its girth and height is the best model. (AIC_GH)")
}else if(min(AIC_GH,AIC_G,AIC_H)==AIC_G){
  print("Predicting the volume from both its girth is the best model. (AIC_G)")
}else if(min(AIC_GH,AIC_G,AIC_H)==AIC_H){
  print("Predicting the volume from both its height is the best model. (AIC_H)")
}
```

```
## [1] "Predicting the volume from both its girth and height is the best model. (AIC_GH)"
```

```
# (c)
library(boot)
getRegr <- function(dat, idx) {
  bsFit <- lm(vec_V ~ vec_G+vec_H, subset=idx, data=dat)
  coef(bsFit)
}
bsRegr <- boot(trees, statistic=getRegr, R=1000)
print(bsRegr)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = trees, statistic = getRegr, R = 1000)
##
##
## Bootstrap Statistics :
##        original      bias    std. error
## t1* -57.9876589 -0.73745219  10.6944549
## t2*   4.7081605 -0.01337318   0.2928448
## t3*   0.3392512  0.01071021   0.1359857
```

```
## intercept, index=1
intercept_CI <- c(boot.ci(bsRegr, conf=0.95, type="bca", index=1)$bca[4],
                  boot.ci(bsRegr, conf=0.95, type="bca", index=1)$bca[5])
print(paste0("The 95% CI for intercept is ", round(intercept_CI[1],4), "~",round(intercept_CI[2],4)))
```

```
## [1] "The 95% CI for intercept is -81.9465~-38.8877"
```

```
## girth, index=2
girth_CI <- c(boot.ci(bsRegr, conf=0.95, type="bca", index=2)$bca[4],
              boot.ci(bsRegr, conf=0.95, type="bca", index=2)$bca[5])
print(paste0("The 95% CI for girth is ", round(girth_CI[1],4), "~",round(girth_CI[2],4)))
```

```
## [1] "The 95% CI for girth is 4.1269~5.2397"
```

```
## height, index=2
height_CI <- c(boot.ci(bsRegr, conf=0.95, type="bca", index=3)$bca[4],
               boot.ci(bsRegr, conf=0.95, type="bca", index=3)$bca[5])
print(paste0("The 95% CI for height is ", round(height_CI[1],4), "~",round(height_CI[2],4)))
```

```
## [1] "The 95% CI for height is 0.0595~0.6133"
```

3

Load the dataset beavers

(a) The last column beaver1$activ is a binary response variable.

Perform a logistic regression to predict activity from day, time, and temperature.

(b) Find out if there is a better predictor with less variables than all 3 combined.

How can you tell if this is a better predictor?

```
rm(list=ls())
data(beavers)
# (a)
beaver1_glm_DayTimeTemp <- glm(activ ~ day+time+temp, family = binomial(link = 'logit'), data = beaver1)
print(beaver1_glm_DayTimeTemp)
```

```
##
## Call:  glm(formula = activ ~ day + time + temp, family = binomial(link = "logit"),
##     data = beaver1)
##
## Coefficients:
## (Intercept)          day         time         temp
##   1.250e+03   -7.182e+00   -6.292e-03    3.354e+01
##
## Degrees of Freedom: 113 Total (i.e. Null);  110 Residual
## Null Deviance:       47.01
## Residual Deviance: 18.2  AIC: 26.2
```

```
# (b)
## 2 factors
beaver1_glm_DayTime <- glm(activ ~ day+time, family = binomial(link = 'logit'), data = beaver1)
beaver1_glm_DayTemp <- glm(activ ~ day+temp, family = binomial(link = 'logit'), data = beaver1)
beaver1_glm_TimeTemp <- glm(activ ~ time+temp, family = binomial(link = 'logit'), data = beaver1)
## 1 factor
beaver1_glm_Day <- glm(activ ~ day, family = binomial(link = 'logit'), data = beaver1)
beaver1_glm_Time <- glm(activ ~ time, family = binomial(link = 'logit'), data = beaver1)
beaver1_glm_Temp <- glm(activ ~ temp, family = binomial(link = 'logit'), data = beaver1)
## We can use AIC to tell which is a better predictor.
## AIC is a relative measure of model fit. It is used for model selection.
## Lower value of AIC suggests a better model.
beaver1_AICs <- c(beaver1_glm_DayTimeTemp$aic,
                  beaver1_glm_DayTime$aic,
                  beaver1_glm_DayTemp$aic,
                  beaver1_glm_TimeTemp$aic,
                  beaver1_glm_Day$aic,
                  beaver1_glm_Time$aic,
                  beaver1_glm_Temp$aic)
beaver1_glm_TimeTemp$aic==min(beaver1_AICs)
```

```
## [1] TRUE
```

```
print("Predicting activity from time and temperature is the best model among these models")
```

```
## [1] "Predicting activity from time and temperature is the best model among these models"
```

4

What does the F-statistic tell you?

When we run an ANOVA test or regression analysis, we get the F statistic value from the result. In general, an F-statistic is a ratio of two quantities.

Under the null hypothesis, the two quantities are expected to be roughly equal, which produces an F-statistic of approximately 1.

On the other hand, if the two quantities are significantly different, we expect F-statistic to be far way from 1.

We can tell whether the two quantities are significantly different by checking if the p value is smaller than 0.05.

5

What does the AIC inform on?

We may always get a closer prediction to the data we have when increasing the number of parameters. However, we may risk overfitting the data. As a result, we use AIC to estimate the models.

Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models.

AIC checks if incresing the number of the parameters really efficiently increases our prediction.

AIC is calculated by “AIC = -2(log-likelihood) + 2K”,

where K is the number of model parameters (the number of variables in the model plus the intercept)

and Log-likelihood is a measure of model fit.

Thus, a lower value of AIC suggests a better model.

AIC helps us deal with both the risk of overfitting and the risk of underfitting.

6

How is logistic regression an extension of linear regression?

Logistic regression is an extension of simple linear regression.

The essential difference between these two is that Logistic regression is used when the dependent variable is binary or categorical in nature.

In contrast, Linear regression is used when the dependent variable is continuous and nature of the regression line is linear.

7 BONUS

Why is it commonplace to take the sum of squared residuals instead of just the residual (not squared)? What advantage(s) does that have?

Advantage 1

If we use the square of the residuals, the residuals below the fitted line (which are negative), would still have to be able to be added up to the positive residuals.

Otherwise, we could have an error of 0 simply because a huge positive residuals could cancel with a huge negative residuals.

Advantage 2

When we square it, instead of just taking the absolute value, we can also have extra penalty for larger residuals.

For example, instead of 2 being 2 times the error of 1, it is 4 times the error of 1 when we square it.

Advantage 3

After we square them, it is mathematically easier to add up all the values when writing programs.

Advantage 4

According the information I found online,

"

from Huber, Robust Statistics, p.10

Two time-honored measures of scatter are the mean absolute deviation [dn = (1/n) * (∑(abd(xi-x_bar)))]

and the mean square deviation [Sn = sqrt((1/n) * (∑((xi-x_bar)**2)))]

There was a dispute between Eddington (1914, p.147) and Fisher (1920, footnote on p. 762) about the relative merits of dn and sn.[…] Fisher seemingly settled the matter by pointing out that for normal observations sn is about 12% more efficient than dn.

"