Project 2: Modeling, Testing, and Predicting

SDS348

05/01/2020

Contents

MaterDolorosa Osueke (mco638)

Modeling

Instructions

A knitted R Markdown document (as a PDF) and the raw R Markdown file (as .Rmd) should both be submitted to Canvas by 11:59pm on 5/1/2020. These two documents will be graded jointly, so they must be consistent (i.e., don't change the R Markdown file without also updating the knitted document). Knit an html copy too, for later! In the .Rmd file for Project 2, you can copy the first code-chunk into your project .Rmd file to get better formatting. Notice that you can adjust the opts_chunk$set(. . .) above to set certain parameters if necessary to make the knitting cleaner (you can globally set the size of all plots, etc). You can copy the set-up chunk in Project2.Rmd: I have gone ahead and set a few for you (such as disabling warnings and package-loading messges when knitting)!

Like before, I envision your written text forming something of a narrative structure around your code/output. All results presented must have corresponding code. Any answers/results/plots etc. given without the corresponding R code that generated the result will not be graded. Furthermore, all code contained in your final project document should work properly. Please do not include any extraneous code or code which produces error messages. (Code which produces warnings is acceptable, as long as you understand what the warnings mean).

Find data:

Find one dataset with at least 5 variables that wish to use to build models. At least one should be categorical (with 2-5 groups) and at least two should be numeric. Ideally, one of your variables will be binary (if not, you will need to create one by discretizing a numeric, which is workable but less than ideal). You will need a minimum of 40 observations (*at least* 10 observations for every explanatory variable you have, ideally 20+ observations/variable).

It is perfectly fine to use either dataset (or the merged dataset, or a subset of your variables) from Project 1. However, you could also diversify your portfolio a bit by choosing a different dataset to work with (particularly if the variables did not reveal interesting associations in Project 1). The only requirement/restriction is that you may not use data from any examples we have done in class or lab. It would be a good idea to pick more cohesive data this time around (i.e., variables that you actually thing might have a relationship you would want to test). Think more along the lines of your Biostats project.

Again, you can use data from anywhere you want (see bottom for resources)! If you want a quick way to see whether a built-in (R) dataset has binary and/or character (i.e., categorical) variables, check out this list: https://vincentarelbundock.github.io/Rdatasets/datasets.html.

Guidelines and Rubric

- **0. (5 pts)** Introduce your dataset and each of your variables (or just your main variables if you have lots) in a paragraph. What are they measuring? How many observations?

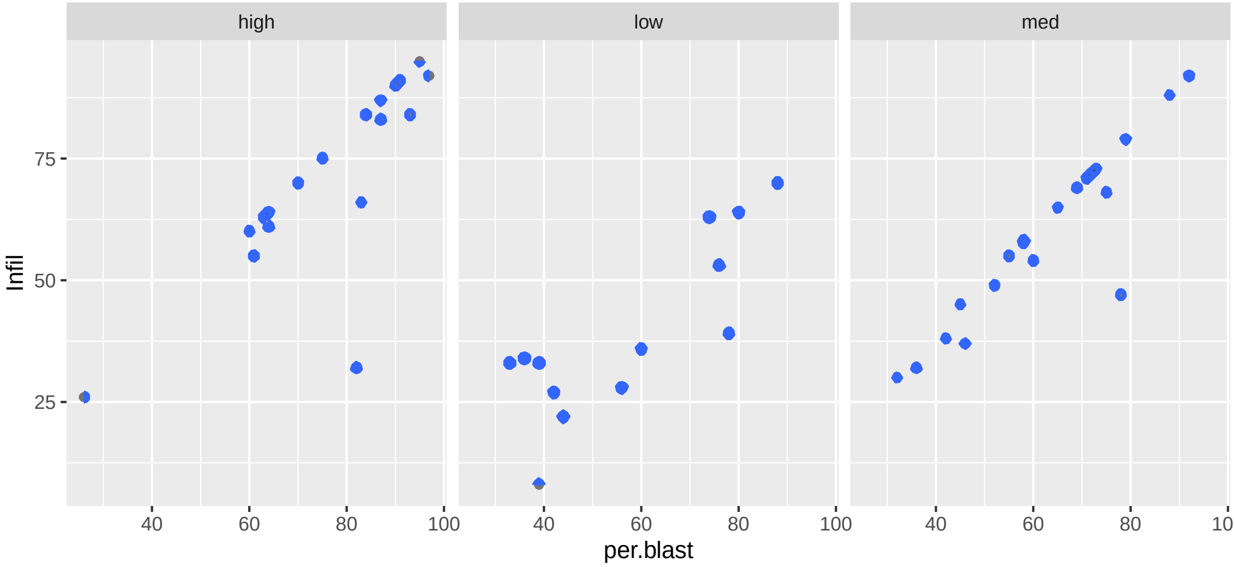
```
library(tidyverse)
library(tidyr)
library(dplyr)
library(devtools)
library(ggplot2)
library(readxl)
Leuk<-read.csv("3Leuk.csv")
head(Leuk)
```

```
##   Infil Index per.blast Time num.blast
## 1    34     5        36    1       low
## 2    63     4        74    2       low
## 3     8     8        39    4       low
## 4    27     5        42    1       low
## 5    22     6        44    1       low
## 6    53    12        76    1       low
```

The dataset that I have decided to use for this project is the response to Treatment for Leukemia or Leuk dataset for short. This data contains information on treatment results for Leukemia patients. 51 untreated patients with acute myeloblastic Leukemia were used and given a course of treatment. Their response to the treatment was assessed and recorded. The data frame contains 51 observations on 5 variables. The variables are Infil, Index, per.blast, Time, and num.blast. Infil describes the percentage of absolute marrow Leukemia infiltrate, Index describes percentage labeling index of the bone marrow leukemia cells, per.blast describes differential percentage of blasts, and num.blast describes the absolute number of blasts, in thousands. The information in this data is important for assessing the effectiveness/effects of certain treatments in order to improve treatments for those suffering with Leukemia.

- **1. (15 pts)** Perform a MANOVA testing whether any of your numeric variables (or a subset of them, if including them all doesn't make sense) show a mean difference across levels of one of your categorical variables (3). If they do, perform univariate ANOVAs to find response(s) showing a mean difference across groups (3), and perform post-hoc t tests to find which groups differ (3). Discuss the number of tests you have performed, calculate the probability of at least one type I error (if unadjusted), and adjust the significance level accordingly (bonferroni correction) before discussing significant differences (3). Briefly discuss assumptions and whether or not they are likely to have been met (2).

```
data(package = .packages(all.available = TRUE))
library(tidyverse)
library(tidyr)
library(dplyr)
library(devtools)
library(ggplot2)
library(readxl)

ggplot(Leuk, aes(x = per.blast, y = Infil)) +
geom_point(alpha = .5) + geom_density_2d(h=2) + coord_fixed() + facet_wrap(~num.blast)
```



```
Covmats<-Leuk%>%group_by(num.blast)%>%do(covs=cov(.[2:3]))
for(i in 1:3){print(as.character(Covmats$num.blast[i])); print(Covmats$covs[i])}
```

```
## [1] "high"
## [[1]]
##                Index per.blast
## Index      27.66340 -15.49673
## per.blast -15.49673 316.30065
##
## [1] "low"
## [[1]]
##               Index per.blast
## Index      7.25641  21.04487
## per.blast 21.04487 387.39744
##
## [1] "med"
## [[1]]
##                 Index  per.blast
## Index     20.302632  -9.342105
## per.blast -9.342105 281.800000
```

```
man1<-manova(cbind(per.blast,Infil)~num.blast, data=Leuk)
summary(man1)
```

```
## Df Pillai approx F num Df den Df Pr(>F)
## num.blast 2 0.45664 7.101 4 96 4.728e-05 ***
## Residuals 48
## ---
## Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1
’ ’ 1
```

```
summary.aov(man1)
```

```
## Response per.blast :
## Df Sum Sq Mean Sq F value Pr(>F)
## num.blast 2 3136.7 1568.37 4.8948 0.01163 *
## Residuals 48 15380.1 320.42
## ---
## Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1
’ ’ 1
##
## Response Infil :
## Df Sum Sq Mean Sq F value Pr(>F)
## num.blast 2 7639.7 3819.9 10.975 0.0001188 ***
## Residuals 48 16706.3 348.0
## ---
## Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1
’ ’ 1
```

```
Leuk%>%group_by(num.blast)%>%summarize(mean(per.blast),mean(Infil))
```

```
## # A tibble: 3 x 3
##   num.blast ‘mean(per.blast)‘ ‘mean(Infil)‘
##   <fct>                 <dbl>         <dbl>
## 1 high                   76.2          71
## 2 low                    57.3          39.2
## 3 med                    62.3          59
```

```
pairwise.t.test(Leuk$per.blast,Leuk$num.blast, p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  Leuk$per.blast and Leuk$num.blast
##
##     high   low
## low 0.0056 -
## med 0.0206 0.4376
##
## P value adjustment method: none
```

```
pairwise.t.test(Leuk$Infil,Leuk$num.blast, p.adj="none")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  Leuk$Infil and Leuk$num.blast
##
##     high    low
## low 2.4e-05 -
## med 0.0535  0.0046
##
## P value adjustment method: none
```

```
1-.95^9
```

```
## [1] 0.3697506
```

```
#Bonferoni
0.05/9
```

```
## [1] 0.005555556
```

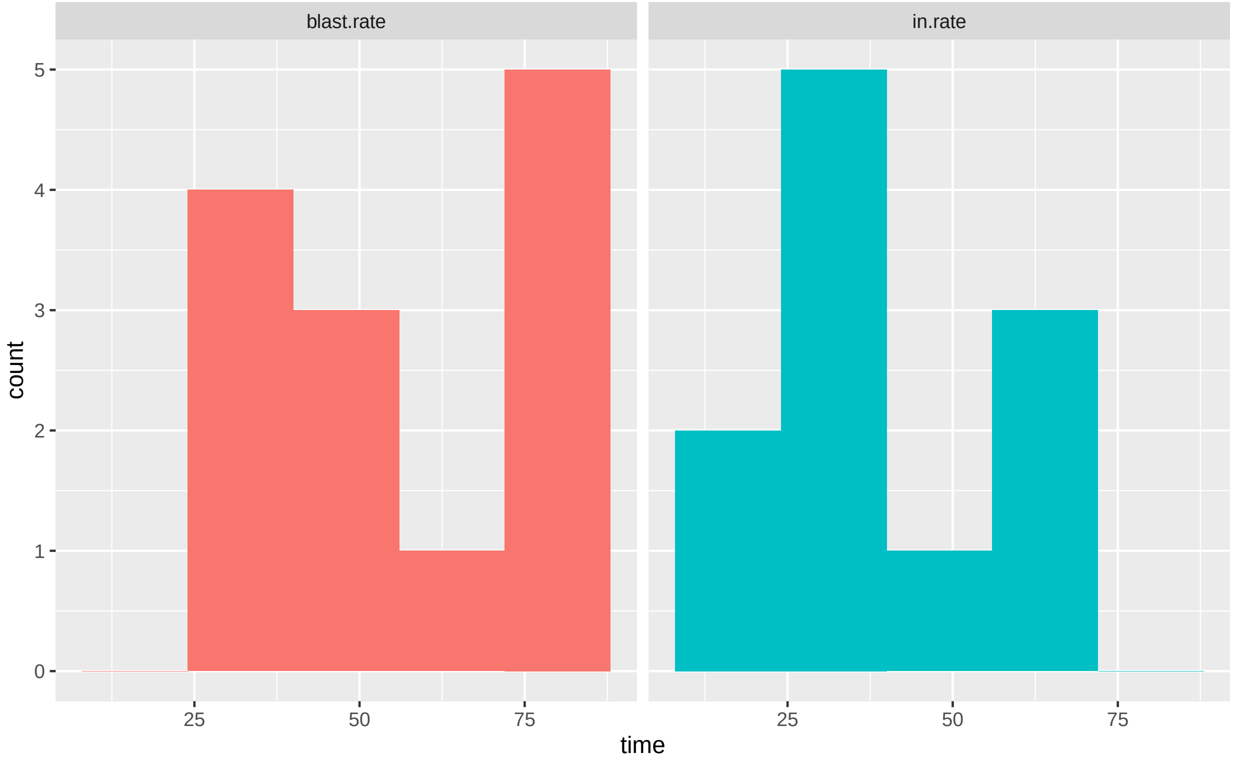
Based on the p-value, the performed MANOVA test shows significance (reject null hypothesis) meaning there is a mean differeence across levels of the categorical variable. After performing a univariate ANOVA for each numerica variable, they both showed significance for per.blast and Infil, at least one num.blast differs. When performing the T-test comparing per.blast rates between combinations of high, low, and medium number of blasts (num.blast) we see a p-valie lower than 0.05 showing we reject the null. When performing the T-test comparing Infil rates between high and low number of blasts (num.blast) we see a p-value greater than 0.05. This shows we fail to reject the null hypothesis meaning they show little to no difference. When comparing other combinations of blasts, they show low p-values so we reject the null; they are definetely different. The number of tests perfomred so far is 9. The probability of at leats one type one error is 0.3697506 (37%). After computing bonferroni correction (0.005555556) and assesing the performed T-test for per.blast rates between combinations of high, low, and medium number of blasts, we see a p-value greater than 0.005 between high and low, and low and med meaning they show little to no difference. For Infil, we see this same result between high and med only. A covariance matix as well as bivariate densities were used to test is assumptions were met for the MANOVA. By looking at the results, it is clear that the assumptions were not met. The values from the covariance matix were not equal and the bivariate densities graph did not look normal/circular.

- **2. (10 pts)** Perform some kind of randomization test on your data (that makes sense). This can be anything you want! State null and alternative hypotheses, perform the test, and interpret the results (7). Create a plot visualizing the null distribution and the test statistic (3).

```
in.rate<-c(34, 63,8,27,22,53,70,33,39,64,36)
blast.rate<-c(36,74,39,42,44,76,88,33,78,80,60,56,39)

cric<-data.frame(Leuk=c(rep("in.rate",11),rep("blast.rate",13)),time=c(in.rate,blast.rate))
head(cric)
```

```
##       Leuk time
## 1 in.rate   34
## 2 in.rate   63
## 3 in.rate    8
## 4 in.rate   27
## 5 in.rate   22
## 6 in.rate   53
```

```
ggplot(cric,aes(time,fill=Leuk))+geom_histogram(bins=6.5)+facet_wrap(~Leuk,ncol=2)+theme(legend.positio
```



```
cric%>%group_by(Leuk)%>%
  summarize(means=mean(time))%>%
  summarize(`mean_diff:`=diff(means))
```

```
## # A tibble: 1 x 1
##   `mean_diff:`
##          <dbl>
## 1        -16.5
```

```
head(perm1<-data.frame(Leuk=cric$Leuk,
                       time=sample(cric$time)))
```

```
##       Leuk time
## 1 in.rate   36
## 2 in.rate   39
## 3 in.rate   88
## 4 in.rate   34
## 5 in.rate   74
## 6 in.rate   39
```

```
perm1%>%group_by(Leuk)%>%
  summarize(means=mean(time))%>%
  summarize(‘mean_diff:‘=diff(means))
```

```
## # A tibble: 1 x 1
##   ‘mean_diff:‘
##          <dbl>
## 1        -1.05
```

```
head(perm2<-data.frame(Leuk=cric$Leuk,
                       time=sample(cric$time)))
```

```
##      Leuk time
## 1 in.rate   36
## 2 in.rate   33
## 3 in.rate   88
## 4 in.rate   53
## 5 in.rate    8
## 6 in.rate   33
```

```
perm2%>%group_by(Leuk)%>%
  summarize(means=mean(time))%>%
  summarize(‘mean_diff:‘=diff(means))
```

```
## # A tibble: 1 x 1
##   ‘mean_diff:‘
##          <dbl>
## 1        -7.09
```

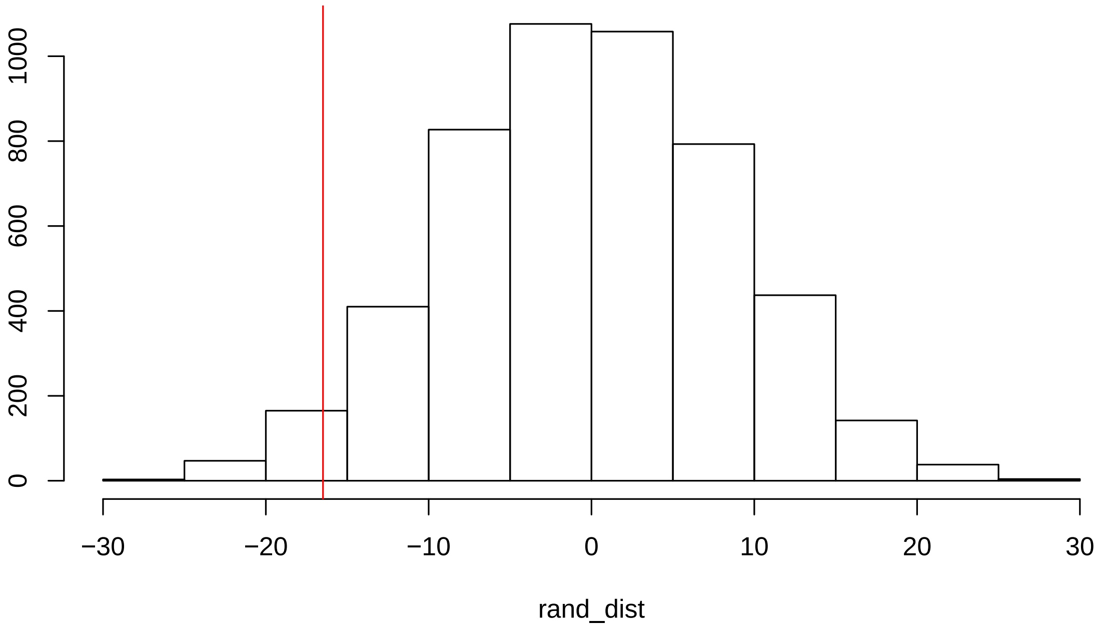
```
head(perm3<-data.frame(Leuk=cric$Leuk,
                       time=sample(cric$time)))
```

```
##      Leuk time
## 1 in.rate   88
## 2 in.rate   22
## 3 in.rate   64
## 4 in.rate   33
## 5 in.rate   60
## 6 in.rate   74
```

```
perm3%>%group_by(Leuk)%>%
  summarize(means=mean(time))%>%
  summarize(‘mean_diff:‘=diff(means))
```

```
## # A tibble: 1 x 1
##   ‘mean_diff:‘
##          <dbl>
## 1        -1.22
```

```r
rand_dist<-vector()

for(i in 1:5000){
new<-data.frame(time=sample(cric$time),Leuk=cric$Leuk)
rand_dist[i]<-mean(new[new$Leuk=="in.rate",]$time)-
              mean(new[new$Leuk=="blast.rate",]$time)}

{hist(rand_dist,main="",ylab=""); abline(v = -16.48951,col="red")}
```



```r
mean(rand_dist>16.48951)*2
```

```
## [1] 0.0504
```

The null hyothesis for the randomization test performed shows there is no assiciation between in.rate and blast.rate by time (or number of blasts on a person). The alternative hypothesis for the randomization test perfomed is that there is an assosication.Based on the randomization test perfomed, We see what the distibution would look like if there was no assiciation with in.rate and blast.rate. It shows what it looks like if the null hypothesis were true. Based on the p-value (0.45) it shows we can reject the null hypothesis.The p-value from the t-test shows a value of 0.051 which shows we can't reject the null hypothesis.

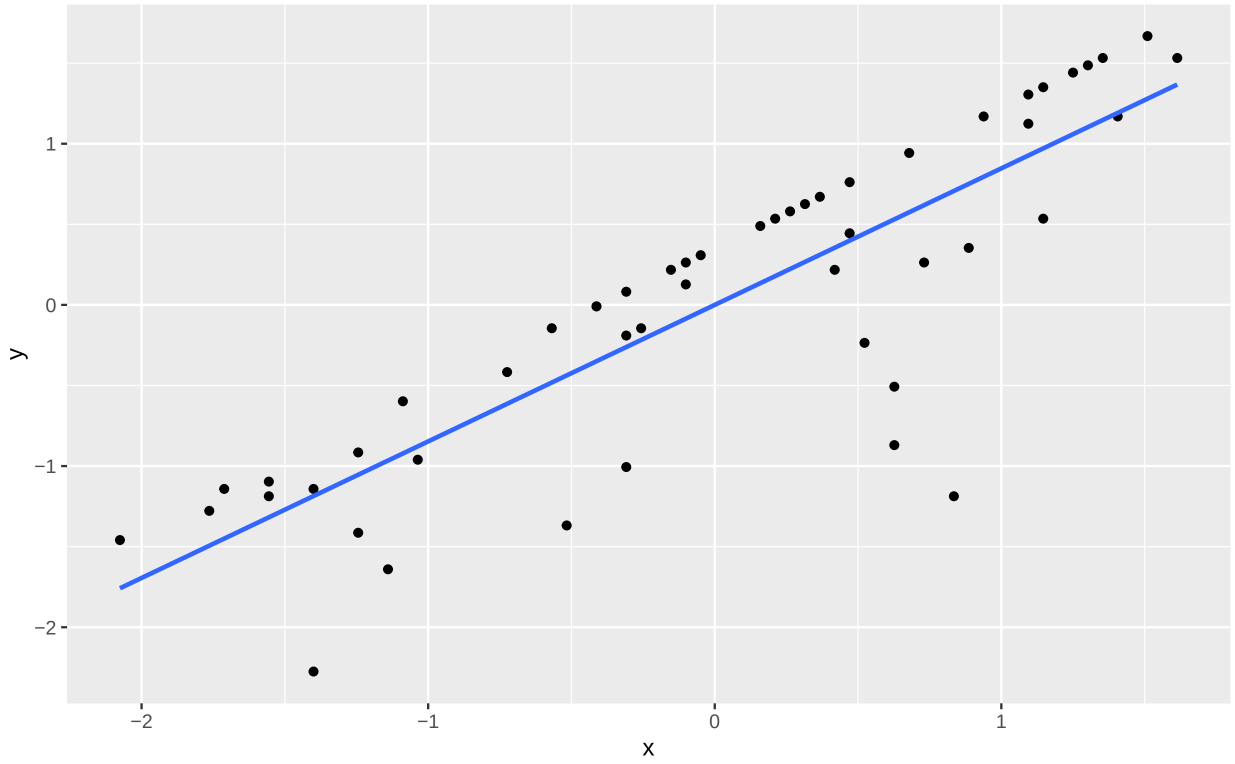
- **3. (35 pts)** Build a linear regression model predicting one of your response variables from at least 2 other variables, including their interaction. Mean-center any numeric variables involved in the interaction.
  - Interpret the coefficient estimates (do not discuss significance) (10)

- Plot the regression using `ggplot()`. If your interaction is numeric by numeric, refer to code near the end of WS15 to make the plot. If you have 3 or more predictors, just chose two to plot for convenience. (8)
- Check assumptions of linearity, normality, and homoskedasticity either graphically or using a hypothesis test (4)
- Regardless, recompute regression results with robust standard errors via `coeftest(..., vcov=vcovHC(...))`. Discuss significance of results, including any changes from before/after robust SEs if applicable. (8)
- What proportion of the variation in the outcome does your model explain? (4)

```r
library(lmtest)
library(sandwich)
x<-scale(Leuk$per.blast)
y<-scale(Leuk$Infil)

lm(y~x)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##  -2.634e-16    8.471e-01
```

```r
ggplot(data.frame(x,y), aes(x,y))+geom_point()+geom_smooth(method="lm",se=F)
```



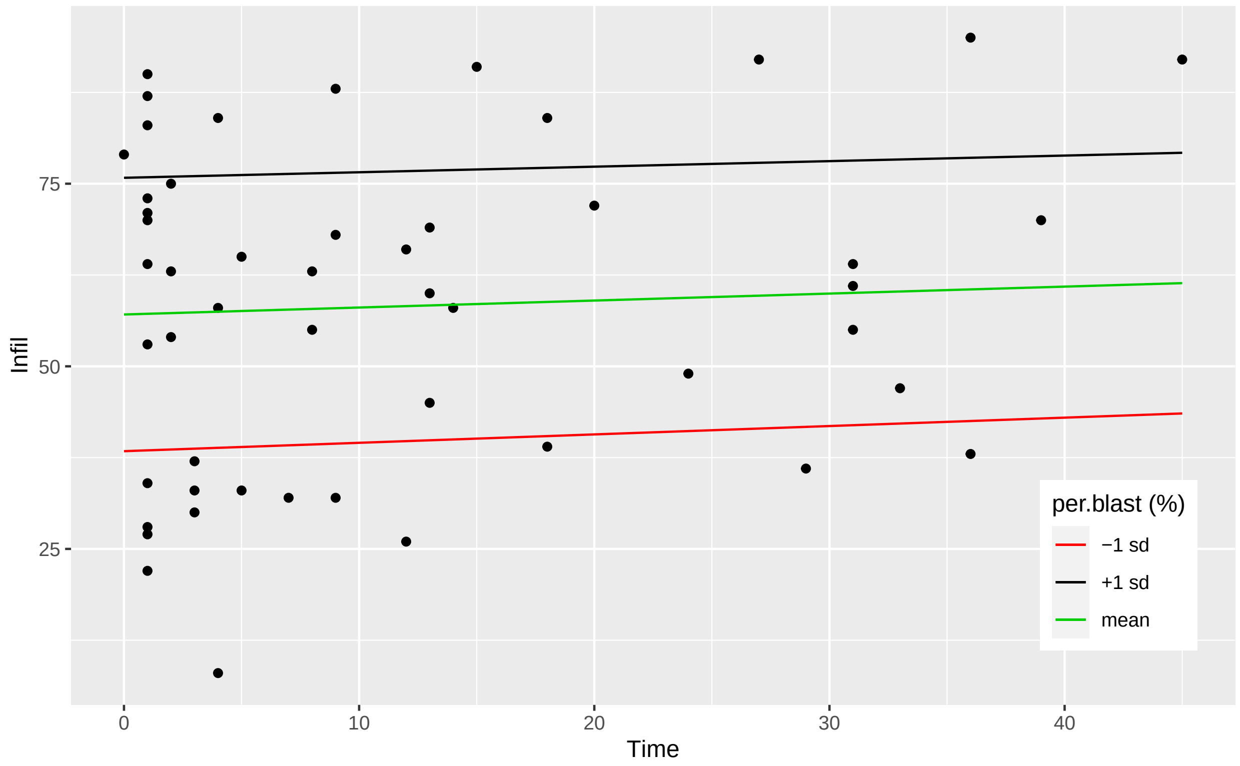
```
fit1<-lm(per.blast~Infil, data=Leuk)
coef(fit1)
```

```
## (Intercept)       Infil
##  22.9465485   0.7387891
```

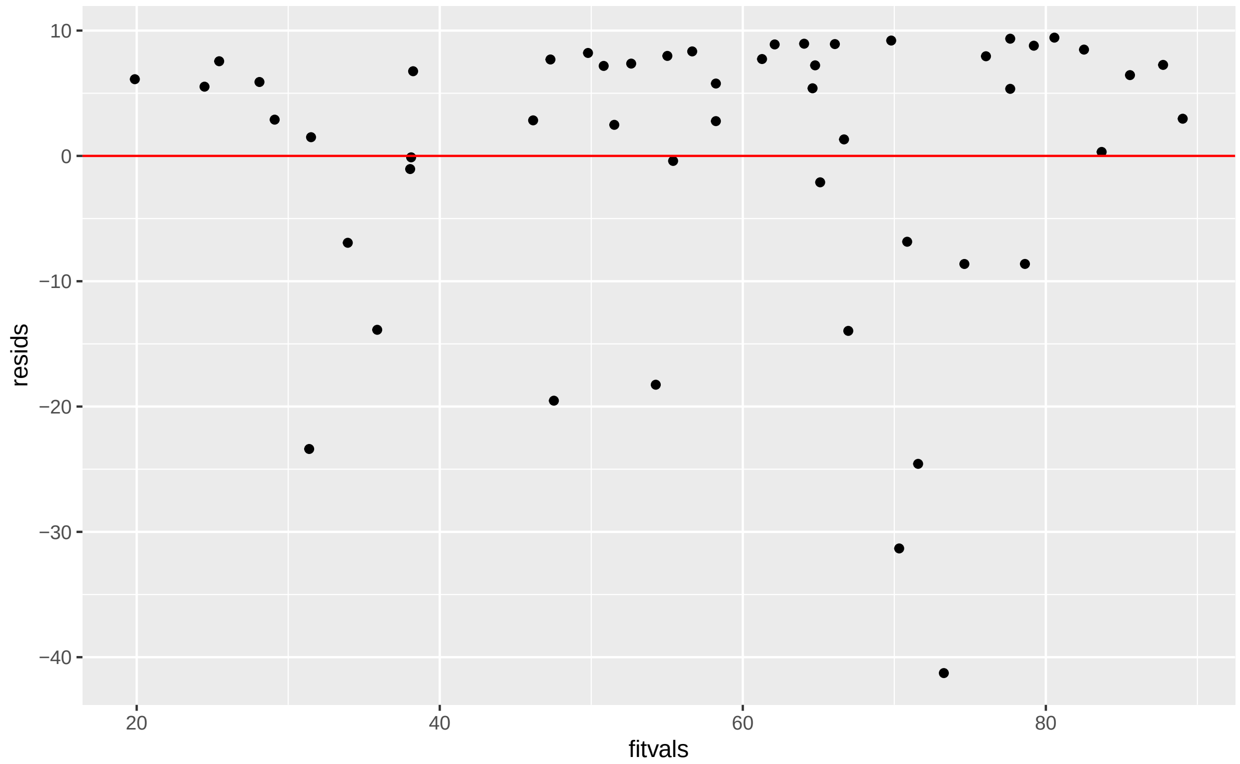
```
summary(fit1)
```

```
##
## Call:
## lm(formula = per.blast ~ Infil, data = Leuk)
##
## Residuals:
## Min 1Q Median 3Q Max
## -16.155 -6.819 -2.580 4.655 35.412
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.9465 4.1153 5.576 1.05e-06 ***
## Infil 0.7388 0.0662 11.159 4.66e-15 ***
## ---
## Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1
’ ’ 1
##
## Residual standard error: 10.33 on 49 degrees of freedom
## Multiple R-squared: 0.7176, Adjusted R-squared: 0.7119
## F-statistic: 124.5 on 1 and 49 DF, p-value: 4.659e-15
```

```
fit2<-lm(Infil ~ per.blast * Time, data=Leuk)
new1<-Leuk
new1$per.blast<-mean(Leuk$per.blast)
new1$mean<-predict(fit2,new1)
new1$per.blast<-mean(Leuk$per.blast)+sd(Leuk$per.blast)
new1$plus.sd<-predict(fit2,new1)
new1$per.blast<-mean(Leuk$per.blast)-sd(Leuk$per.blast)
new1$minus.sd<-predict(fit2,new1)
newint<-new1%>%select(Infil,per.blast,mean,plus.sd,minus.sd)%>%gather(per.blast,value,-Infil)

mycols<-c("#619CFF","#F8766D","#00BA38")
names(mycols)<-c("-1 sd","mean","+1 sd")
mycols=as.factor(mycols)

ggplot(Leuk,aes(Time,Infil),group=mycols)+geom_point()+geom_line(data=new1,aes(y=mean,color="mean"))+ge
```



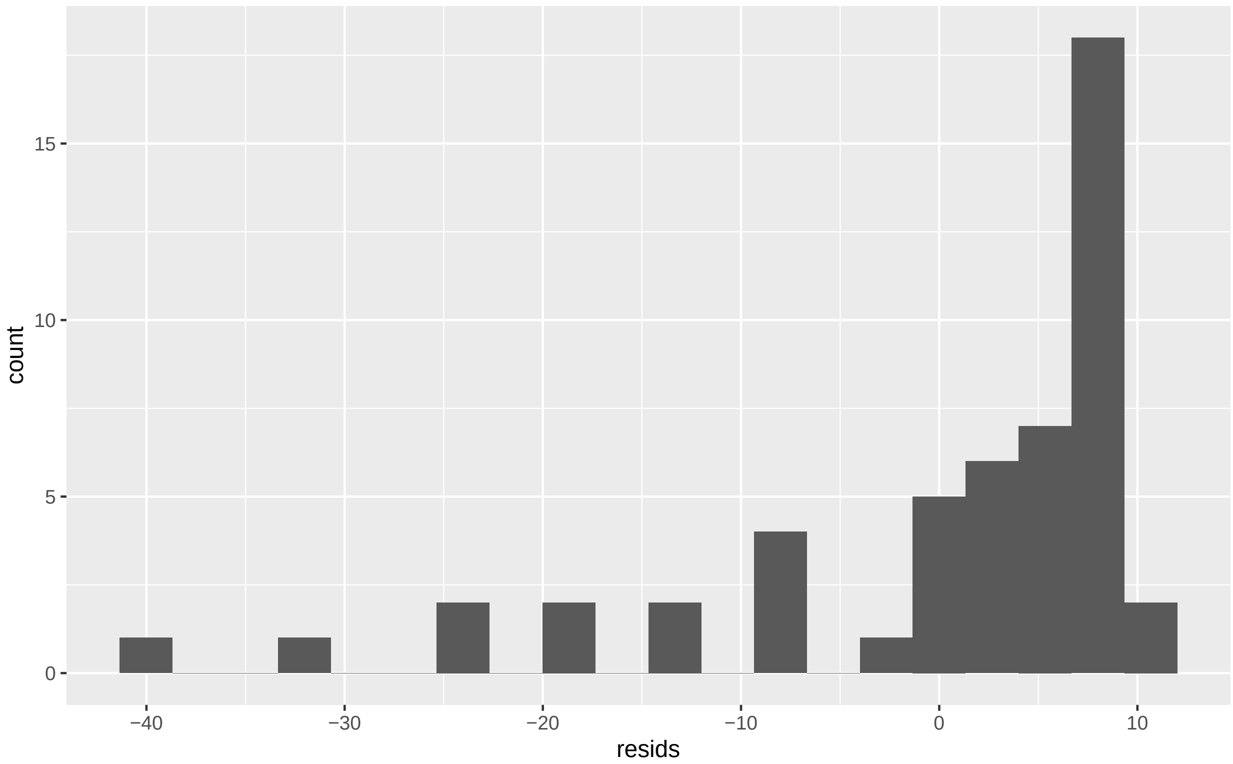
```
#Assumptions
resids<-fit2$residuals
fitvals<-fit2$fitted.values


ggplot()+geom_point(aes(fitvals,resids))+geom_hline(yintercept = 0, col="red")
```



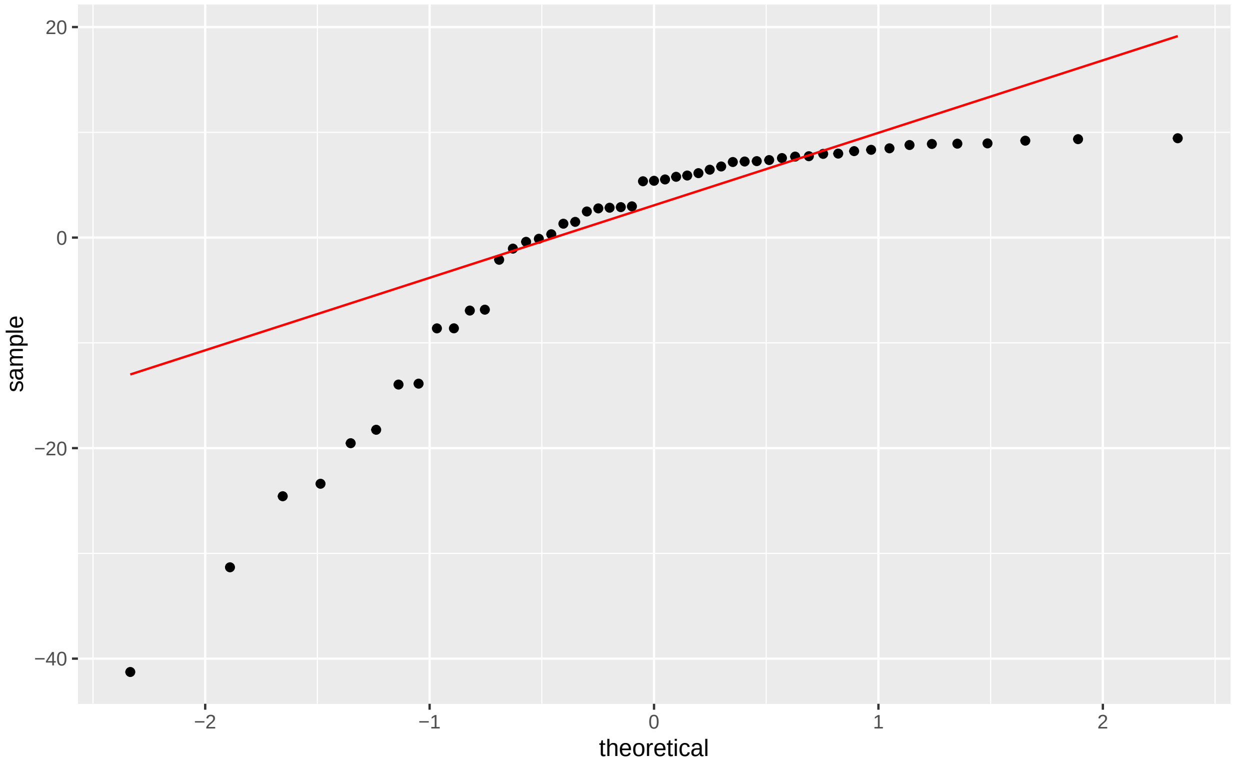
```
bptest(fit2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit2
## BP = 0.7873, df = 3, p-value = 0.8525
```

```
ggplot()+geom_histogram(aes(resids),bins=20)
```



```
ggplot()+geom_qq(aes(sample=resids))+geom_qq_line(aes(sample=resids), color='red')
```



```r
ks.test(resids, "pnorm", sd=sd(resids))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  resids
## D = 0.21176, p-value = 0.01742
## alternative hypothesis: two-sided
```

```r
#uncorrected SE
summary(fit2)$coef[,1:2]
```

```
##                    Estimate  Std. Error
## (Intercept)    -7.031955282 8.200979393
## per.blast       0.972492438 0.120015776
## Time            0.161669733 0.556717809
## per.blast:Time -0.001004849 0.007546648
```

```r
#correctedSE
coeftest(fit2, vcov=vcovHC(fit2))[,1:2]
```

```
##                    Estimate  Std. Error
## (Intercept)    -7.031955282 7.300025263
## per.blast       0.972492438 0.110950250
## Time            0.161669733 0.383451238
## per.blast:Time -0.001004849 0.005500866
```

```r
SST <- sum((Leuk$Infil-mean(Leuk$Infil))^2)
SSR <- sum((fit2$fitted.values-mean(Leuk$Infil))^2)
SSE <- sum(fit2$residuals^2)
SSR/SST
```

```
## [1] 0.7202629
```

```r
summary(fit2)$r.sq
```

```
## [1] 0.7202629
```

The coefficient estimates obtained are ~0 for the intercept and 0.847 for x. This shows the intercept is 0 and the slope is 0.847. When looking at the plot for linearity, the plot did not look as if they meet assumptions.When loiking at the plot for homoskedacity it is not very clear but seems as if homoskedacity was met. When looking at normality, the ggplots also looked as if normality was not met. This was followed up with hypothesis test. Based on the p-value from the Breusch-Pagan test, it can be concluded assumtions for linearity and homoskedasticity was met and not violated. It showed a p-value of 0.8525. The p-vaule on the One-sample Kolmogorov-Smirnov test performed which was less than 0.05 showing the assumption for normality was not met and violated. After computing uncorrected and corrrected SE with coeftest we see that the estimates remained the same before and after however, the robust standard errors got slightly smaller which is odd. However the difference was not extreme. The proportion of the variation in the outcome of the model was 0.7202629.

- **4. (5 pts)** Rerun same regression model (with interaction), but this time compute bootstrapped standard errors. Discuss any changes you observe in SEs and p-values using these SEs compared to the original SEs and the robust SEs)

```
fit1<-lm(per.blast~Infil, data=Leuk)
summary(fit1)
```

```
##
## Call:
## lm(formula = per.blast ~ Infil, data = Leuk)
##
## Residuals:
## Min 1Q Median 3Q Max
## -16.155 -6.819 -2.580 4.655 35.412
##
## Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.9465 4.1153 5.576 1.05e-06 ***
## Infil 0.7388 0.0662 11.159 4.66e-15 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
' ' 1
##
## Residual standard error: 10.33 on 49 degrees of freedom
## Multiple R-squared: 0.7176, Adjusted R-squared: 0.7119
## F-statistic: 124.5 on 1 and 49 DF, p-value: 4.659e-15
```

```
coeftest(fit2)[,1:2]
```

```
##                    Estimate  Std. Error
## (Intercept)    -7.031955282 8.200979393
## per.blast       0.972492438 0.120015776
## Time            0.161669733 0.556717809
## per.blast:Time -0.001004849 0.007546648
```

```
coeftest(fit2, vcov=vcovHC(fit2))[,1:2]
```

```
##                    Estimate  Std. Error
## (Intercept)    -7.031955282 7.300025263
## per.blast       0.972492438 0.110950250
## Time            0.161669733 0.383451238
## per.blast:Time -0.001004849 0.005500866
```

```
samp_distn<-replicate(5000,{
  boot_dat<-boot_dat<-Leuk[sample(nrow(Leuk),replace = TRUE),]
  fit2<-lm(Infil ~ per.blast * Time, data=Leuk)
coef(fit2)
})
samp_distn%>%t%>%as.data.frame%>%summarize_all(sd)
```

```
##   (Intercept) per.blast Time per.blast:Time
## 1           0         0    0              0
```

```
samp_distn%>%t%>%as.data.frame%>%summarize_all(sd)
```

```
##   (Intercept) per.blast Time per.blast:Time
## 1           0         0    0              0
```

```
fit4<-lm(Infil ~ per.blast * Time, data=Leuk)
resids<-fit4$residuals
fitted<-fit4$fitted.values

resid_resamp<-replicate(5000,{
new_resids<-sample(resids,replace=TRUE)
newdat<-Leuk
newdat$new_y<-fitted+new_resids
fit4<-lm(Infil ~ per.blast * Time, data=Leuk)
coef(fit4)
})

resid_resamp%>%t%>%as.data.frame%>%summarize_all(sd)
```

```
##   (Intercept) per.blast Time per.blast:Time
## 1           0         0    0              0
```

```
coeftest(fit4)[,1:2]
```

```
##                    Estimate  Std. Error
## (Intercept)    -7.031955282 8.200979393
## per.blast       0.972492438 0.120015776
## Time            0.161669733 0.556717809
## per.blast:Time -0.001004849 0.007546648
```

```
coeftest(fit4, vcov=vcovHC(fit4))[,1:2]
```

```
##                    Estimate  Std. Error
## (Intercept)    -7.031955282 7.300025263
## per.blast       0.972492438 0.110950250
## Time            0.161669733 0.383451238
## per.blast:Time -0.001004849 0.005500866
```

```
samp_distn%>%t%>%as.data.frame%>%summarize_all(sd)
```

```
##   (Intercept) per.blast Time per.blast:Time
## 1           0         0    0              0
```

```
resid_resamp%>%t%>%as.data.frame%>%summarize_all(sd)
```

```
##   (Intercept) per.blast Time per.blast:Time
## 1           0         0    0              0
```

Upon rerunning the regression model and analyzing the bootstrap data, we see and estimated intercept of 0, an estimated per.blast standard error of 0, and estimated Time standard error of 0. This is a bit strange to have 0 for the standard error but this is much lower than the robust and original standard errors. Because the SE is lower than the original and robust it will have a higher pvalue.

- **5. (40 pts)** Perform a logistic regression predicting a binary categorical variable (if you don't have one, make/get one) from at least two explanatory variables (interaction not necessary).
  - Interpret coefficient estimates in context (10)
  - Report a confusion matrix for your logistic regression (2)
  - Compute and discuss the Accuracy, Sensitivity (TPR), Specificity (TNR), and Recall (PPV) of your model (5)
  - Using ggplot, plot density of log-odds (logit) by your binary outcome variable (3)
  - Generate an ROC curve (plot) and calculate AUC (either manually or with a package); interpret (10)
  - Perform 10-fold (or repeated random sub-sampling) CV and report average out-of-sample Accuracy, Sensitivity, and Recall (10)

```
merg<-Leuk%>%mutate(Time.mute = case_when(Time>11 ~ "1", Time<12 ~ "0"))

head(merg)
```

```
##   Infil Index per.blast Time num.blast Time.mute
## 1    34     5        36    1       low         0
## 2    63     4        74    2       low         0
## 3     8     8        39    4       low         0
## 4    27     5        42    1       low         0
## 5    22     6        44    1       low         0
## 6    53    12        76    1       low         0
```

```
library(tidyverse); library(lmtest)

data<-merg
data$y<-ifelse(merg$Time.mute==1,1,0) #high survival = 1

fit7<-glm(y~Infil+per.blast,data=data,family="binomial")
summary(fit7)
```

```
##
## Call:
## glm(formula = y ~ Infil + per.blast, family =
"binomial", data = data)
##
## Deviance Residuals:
## Min 1Q Median 3Q Max
## -1.2245 -1.0533 -0.8494 1.2646 1.5497
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087820 1.069799 -1.017 0.309
## Infil 0.020337 0.026085 0.780 0.436
## per.blast -0.007026 0.029627 -0.237 0.813
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 69.104 on 50 degrees of freedom
## Residual deviance: 67.752 on 48 degrees of freedom
## AIC: 73.752
##
## Number of Fisher Scoring iterations: 4
```

```
coeftest(fit7)
```

```
##
## z test of coefficients:
##
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.0878202  1.0697995 -1.0168   0.3092
## Infil        0.0203369  0.0260852  0.7796   0.4356
## per.blast   -0.0070264  0.0296268 -0.2372   0.8125
```

```
coef(fit7)
```

```
##  (Intercept)        Infil    per.blast
## -1.087820193  0.020336913 -0.007026383
```

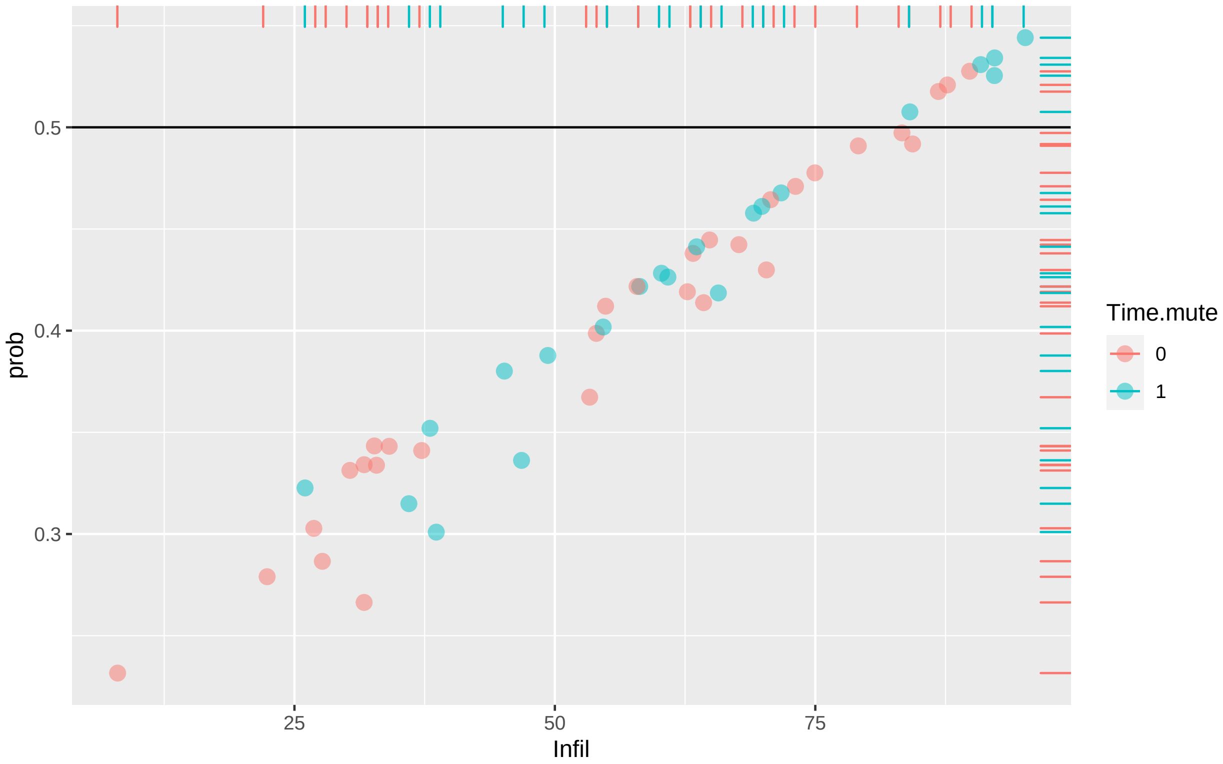
```
exp(coef(fit7))%>%round(3)
```

```
## (Intercept)       Infil   per.blast
##       0.337       1.021       0.993
```

```
coef(fit7)%>%exp%>%round(5)%>%data.frame()
```

```
##                   .
## (Intercept) 0.33695
## Infil       1.02055
## per.blast   0.99300
```

```
data$prob<-predict(fit7,type="response") #save predicted probabilities
data$Time.mute<-as.factor(data$Time.mute)

ggplot(data, aes(Infil,prob))+geom_jitter(aes(color=Time.mute),alpha=.5,size=3)+geom_rug(aes(color=Time
```



```
#matrix
prob<-predict(fit7,type="response")
tab<-table(predict=as.numeric(data$prob>.5),truth=data$y)%>%addmargins
tab
```

```
##         truth
## predict  0  1 Sum
##     0   27 16  43
##     1    3  5   8
##     Sum 30 21  51
```

```
#Sensitivity
5/21
```

```
## [1] 0.2380952
```

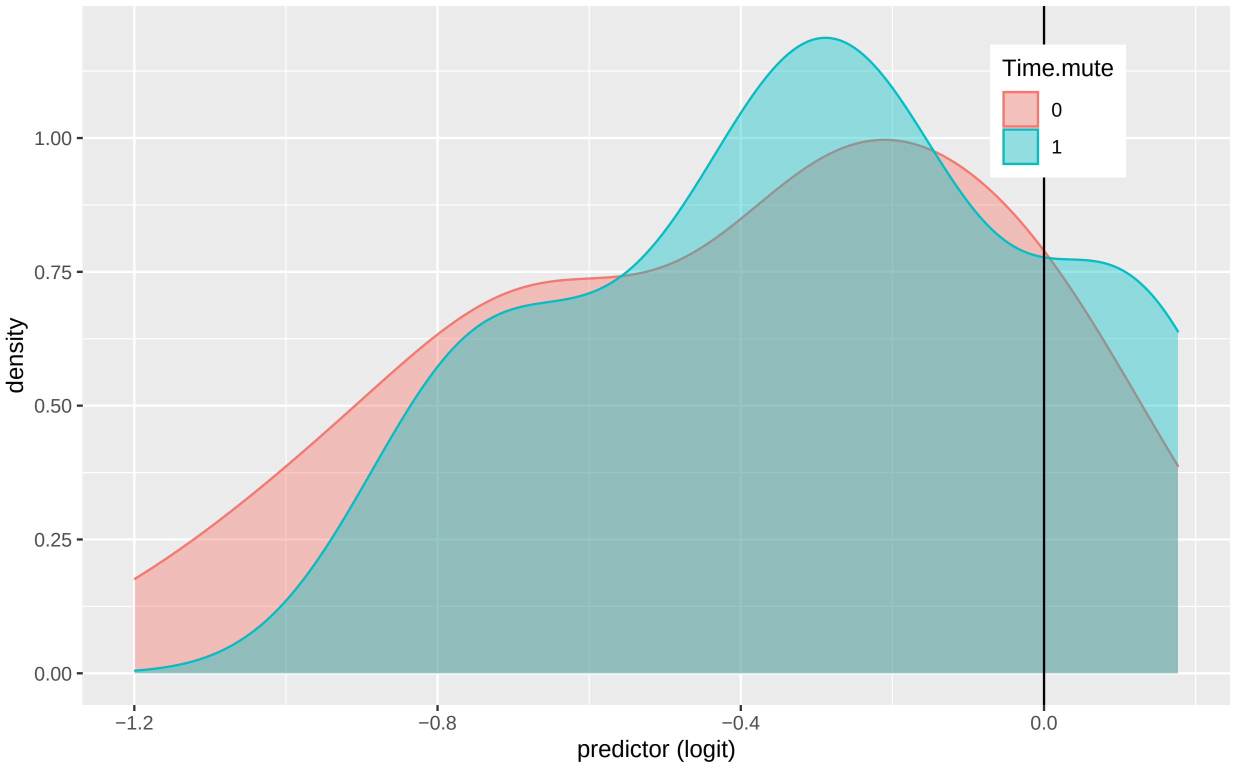
```
#Specificity
25/30
```

```
## [1] 0.8333333
```

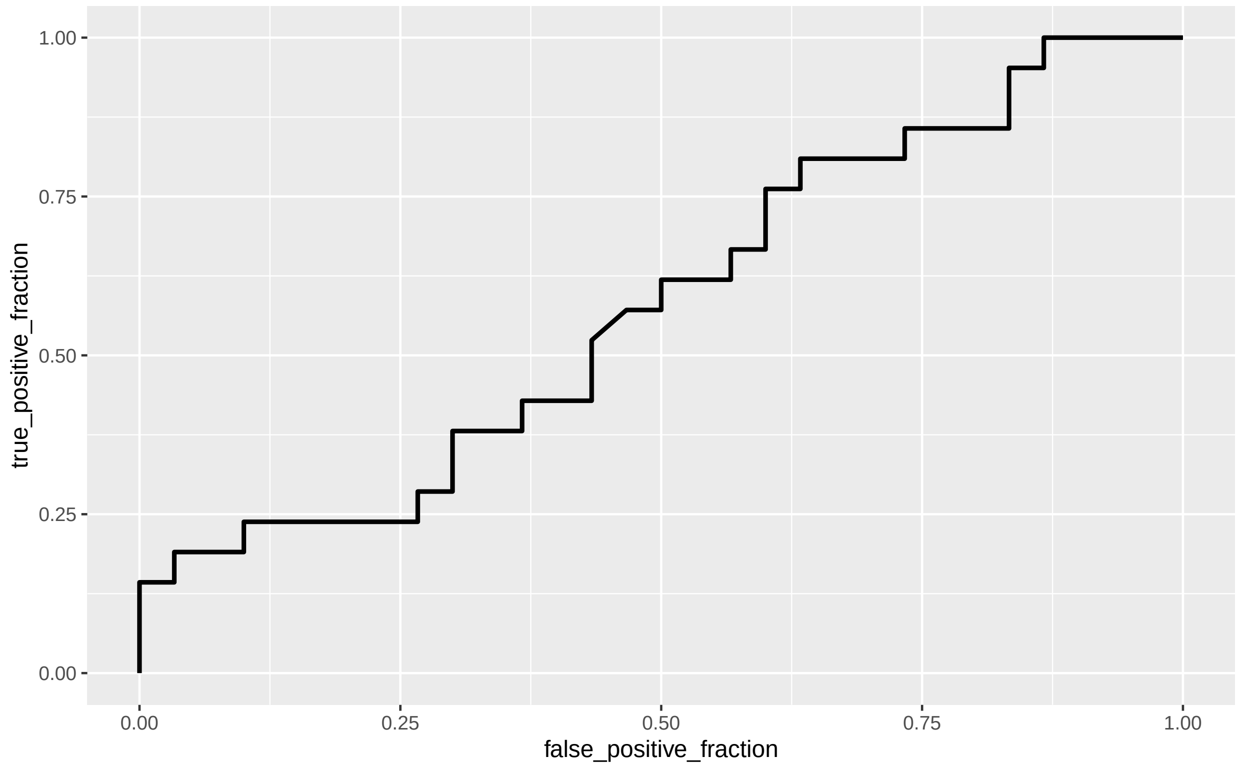
```
#PPV
5/10
```

```
## [1] 0.5
```

```
#Plot
data$logit<-predict(fit7,type="link") #get predicted logit scores (logodds)

data%>%ggplot()+geom_density(aes(logit,color=Time.mute,fill=Time.mute), alpha=.4)+theme(legend.position=
```



```
#ROC
library(plotROC) #install.packages(plotROC)
#geom_roc needs actual outcome (0,1) and predicted prob (or predictor if just one)
ROCplot<-ggplot(data)+geom_roc(aes(d=y,m=prob), n.cuts=0)
ROCplot
```



```r
calc_auc(ROCplot)
```

```
##   PANEL group       AUC
## 1     1    -1 0.5785714
```

```r
#CV
merg<-Leuk%>%mutate(Time.mute = case_when(Time>11 ~ "1", Time<12 ~ "0"))
fit7<-glm(y~Infil+per.blast,data=data,family="binomial")
prob<-predict(fit7,type="response")

class_diag <- function(probs,truth){

  #CONFUSION MATRIX: CALCULATE ACCURACY, TPR, TNR, PPV
  tab<-table(factor(probs>.5,levels=c("FALSE","TRUE")),truth)
  acc=sum(diag(tab))/sum(tab)
  sens=tab[2,2]/colSums(tab)[2]
  spec=tab[1,1]/colSums(tab)[1]
  ppv=tab[2,2]/rowSums(tab)[2]

  if(is.numeric(truth)==FALSE & is.logical(truth)==FALSE) truth<-as.numeric(truth)-1

  #CALCULATE EXACT AUC
  ord<-order(probs, decreasing=TRUE)
  probs <- probs[ord]; truth <- truth[ord]

  TPR=cumsum(truth)/max(1,sum(truth))
  FPR=cumsum(!truth)/max(1,sum(!truth))
  dup<-c(probs[-1]>=probs[-length(probs)], FALSE)
  TPR<-c(0,TPR[!dup],1); FPR<-c(0,FPR[!dup],1)

  n <- length(TPR)
  auc<- sum( ((TPR[-1]+TPR[-n])/2) * (FPR[-1]-FPR[-n]) )

  data.frame(acc,sens,spec,ppv,auc)
}

class_diag(prob,data$y)
```

```
##        acc      sens spec   ppv       auc
## 1 0.627451 0.2380952  0.9 0.625 0.5785714
```

```
#install.packages(pROC)
library(pROC) #Compare with this AUC calculator!
auc(data$y,prob)
```

```
## Area under the curve: 0.5786
```

```
set.seed(1234)
k=10 #choose number of folds

data<-data[sample(nrow(data)),] #randomly order rows
folds<-cut(seq(1:nrow(data)),breaks=k,labels=F) #create folds

diags<-NULL
for(i in 1:k){
  ## Create training and test sets
  train<-data[folds!=i,]
  test<-data[folds==i,]

  truth<-test$y ## Truth labels for fold i

  ## Train model on training set (all but fold i)
  fit7<-glm(y~Infil+per.blast,data=data,family="binomial")

  ## Test model on test set (fold i)
  probs<-predict(fit7,newdata = test,type="response")

  ## Get diagnostics for fold i
  diags<-rbind(diags,class_diag(probs,truth))
}


summarize_all(diags,mean)
```

```
##         acc sens      spec ppv       auc
## 1 0.6266667  0.2 0.9083333 NaN 0.6166667
```

```
mean(diags)
```

```
## [1] NA
```

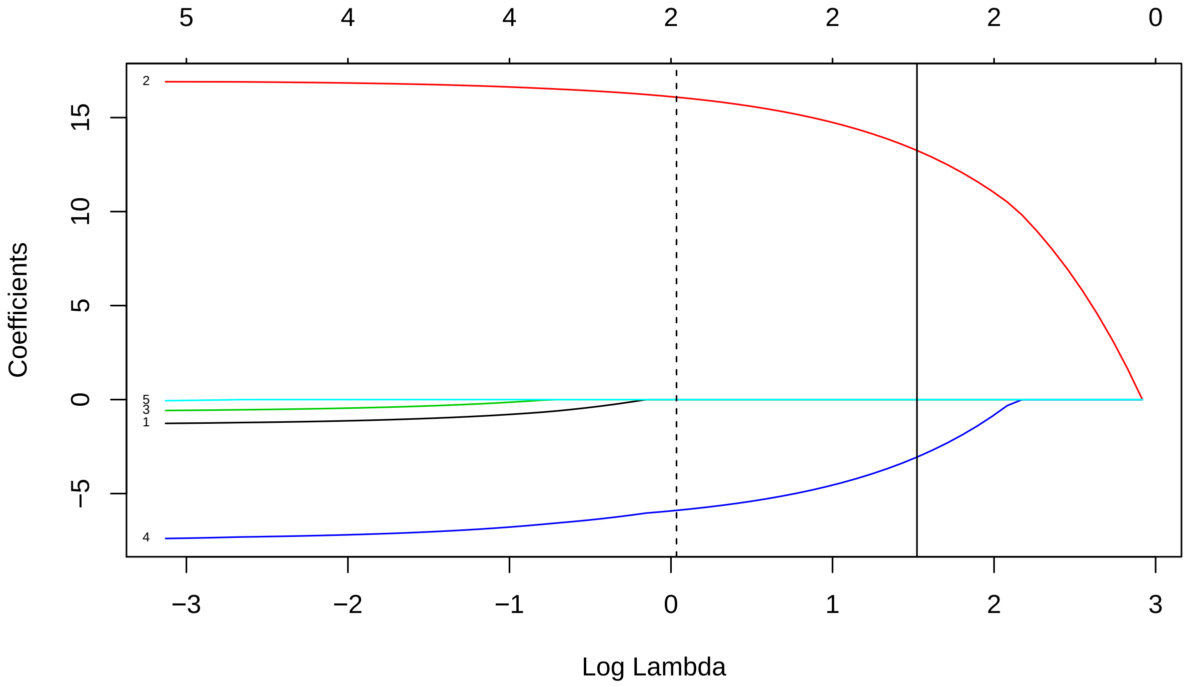
Based on the logistic regression using one categorical variable (Time.mute), the odds scale coefficients show that the intercept is 0.33695 when everything else is equal to 0. We have a value of 1.02055 for infil coefficient which shoes that for every one addition of Infil, that multiplies the odds of survival time(Time.mute) by 1.02055. We have a value of 0.99300 for the per.blast coefficient which shoes that for every one addition/increase of per.blast,that multiplies the odds of survival time(Time.mute) by 0.99300.They all show positive effects on the probability. The Sensitivity for the matrix shows out of all Times that were actually high what proportion were predicted as high survival. This value was 0.2380952. The Specificity or true negative rate turned out to be 0.8333333. The PPV which showed proportion of those classified high survival that actual are in the model was 0.5. The speciificity (true negative) had the highest percentage at 83%. The AUC of the ROC curve generated is 0.5880952. This value is the probability that a randomly selected Leukemia treated patienet with high survival has about half a predicted probability of being high than a randomly selected patient with low survival. After performing the 10-fold CV, the accuracy showed a value of 0.6266667, sensitivity showed a value of 0.35, and recall had a value of "Nan". The AUC value was 0.6625.

- **6. (10 pts)** Choose one variable you want to predict (can be one you used from before; either binary or continuous) and run a LASSO regression inputting all the rest of your variables as predictors. Choose lambda to give the simplest model whose accuracy is near that of the best (i.e., `lambda.1se`). Discuss which variables are retained. Perform 10-fold CV using this model: if response in binary, compare model's out-of-sample accuracy to that of your logistic regression in part 5; if response is numeric, compare the residual standard error (at the bottom of the summary output, aka RMSE): lower is better fit!

```
#Infil
library(glmnet)
y<-as.matrix(Leuk$Infil) #grab response
x<-model.matrix(Infil~.,data=Leuk)[,-1] #grab predictors
head(x)
```

```
##   Index per.blast Time num.blastlow num.blastmed
## 1     5        36    1            1            0
## 2     4        74    2            1            0
## 3     8        39    4            1            0
## 4     5        42    1            1            0
## 5     6        44    1            1            0
## 6    12        76    1            1            0
```

```
x<-scale(x)

cv <- cv.glmnet(x,y)
{plot(cv$glmnet.fit, "lambda", label=TRUE); abline(v = log(cv$lambda.1se)); abline(v = log(cv$lambda.mi
```



```
lasso<-glmnet(x,y,lambda=cv$lambda.1se)
coef(lasso)
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept)  58.196078
## Index          .
## per.blast    13.251488
## Time           .
## num.blastlow -3.060906
## num.blastmed  .
```

```
set.seed(1234)
k=10 #choose number of folds

data1<-Leuk[sample(nrow(Leuk)),] #randomly order rows
folds<-cut(seq(1:nrow(Leuk)),breaks=k,labels=F) #create folds

diags<-NULL
for(i in 1:k){
  train<-data1[folds!=i,]
  test<-data1[folds==i,]

  fit<-lm(Infil~per.blast+num.blast,data=train)
  yhat<-predict(fit,newdata=test)

  diags<-mean((test$Infil-yhat)^2)
}
```

```
mean(diags)
```

```
## [1] 46.6851
```

After runnning LASSO, we see that per.blast and num.blastlow are the only variables retained showing that they are important for predicting percentage of absolute marrow leukemia infiltrate (Infil) in Leukemia patients given treated.After doing cross validation a value of 10.85026 was obtained. When compared to the value in part 5, we see that the value is much lower. In question 5, the value was 32.56109 compared to 10.85026 obtained.This shows a better fit. When conducting the 10-fold CV, the ACC, Sens, spec, ppv, and auc had a similar value to question 5.

```
data(package = .packages(all.available = TRUE))
```

...