Assignment 1

Andy Yuan

2025-01-26

Question 1

a)

$$L(P_{B0}) = (1 - P_{B0})^{20} P_{B0}^{30}$$

$$L(P_{B1}) = (1 - P_{B1})^{20} P_{B1}^{30}$$

$$LRT = -2log\frac{(1 - P_{B0})^{20} P_{B0}^{30}}{(1 - P_{B1})^{20} P_{B1}^{30}}$$

$$= -2[20log(1 - P_{B0}) + 30log(P_{B0}) - 20log(1 - P_{B1}) - 30log(P_{B1})]$$

$$= -2[50log(P_{B0}) - 20log(1 - P_{B1}) - 30log(P_{B1})]$$

$$= -2[50log(0.5) - 20log(0.4) - 30log(0.6)] = 2.013551$$

The null hypothesis is that the two models are the same. The alternative is that the two models differ.

In this case, there are 49 degrees of freedom. The LRT follows a chi-squared distribution. The test statistic of 2.013551 is less than the critical value of 55.758 at $\alpha = 0.05$, so we fail to reject the null hypothesis.

b)

In this case, the $LRT = -2[1000log(P_{B0}) - 400log(1 - P_{B1}) - 600log(P_{B1})]$ on 999 degrees of freedom.

The test statistic is 20.13551, which is less than the critical value of 124.342 at $\alpha = 0.05$ and 100 degrees of freedom.

c)

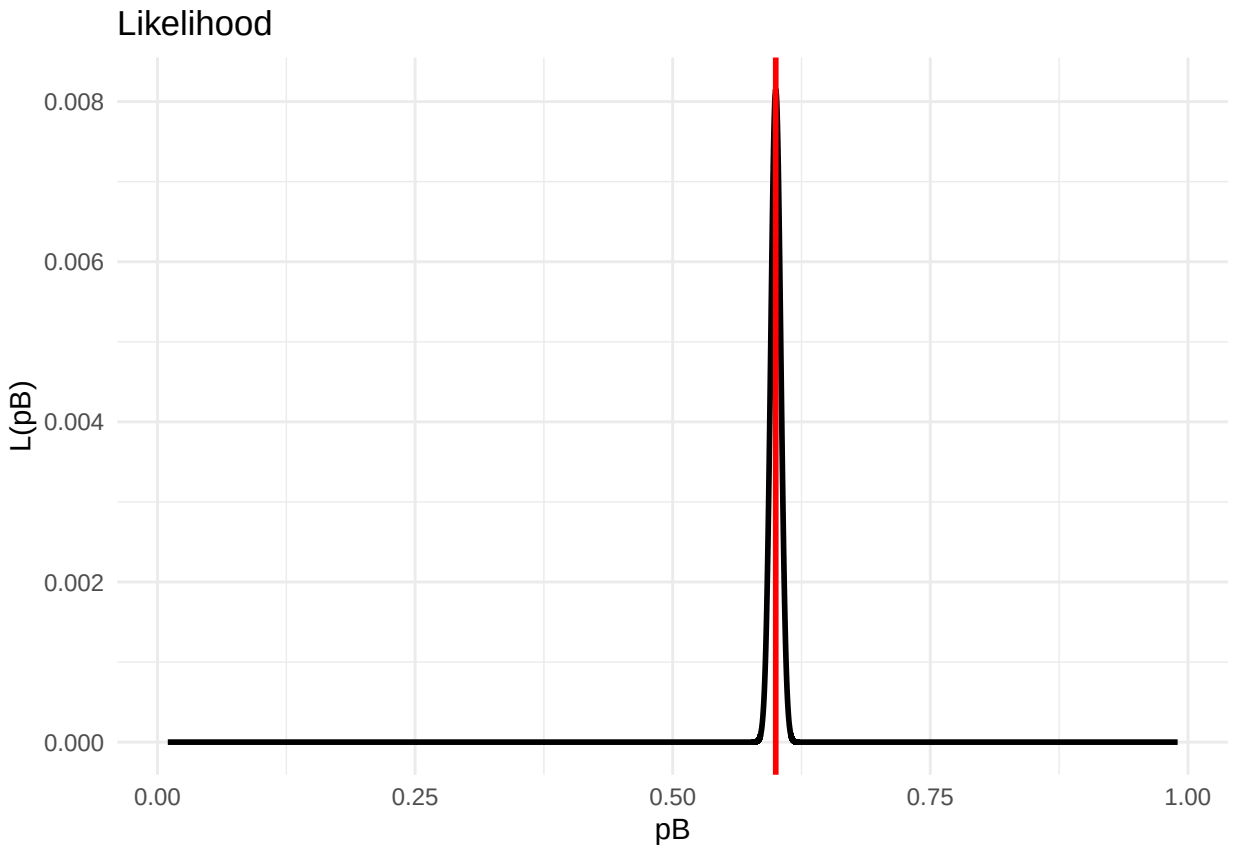
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

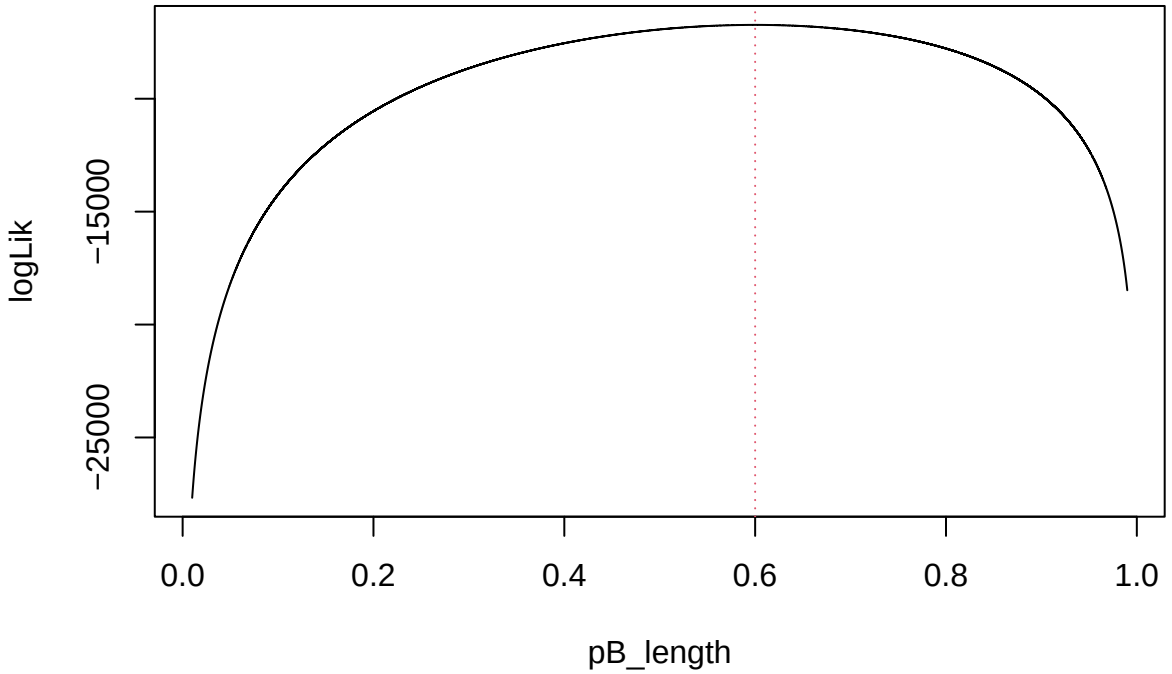
```r
logLik <- function(p,n.boys,n.girls){
Lik=dbinom(n.boys, size = n.boys + n.girls, prob = p)
logLik=n.boys*log(p) + n.girls*log(1-p)
lik.frame <- data.frame(p,Lik,logLik)
return(lik.frame)
}

pB_length=seq(0.01,0.99,length = 100000)
df = logLik(p=pB_length,n.boys=6000,n.girls=4000)
likelihood_plot <- ggplot(data.frame(pB = pB_length, Likelihood = df$Lik),
                          aes(x = pB, y = Likelihood)) +
  geom_line(color= 'black', size = 1) +
  geom_vline(xintercept = 0.6, color = 'red', linetype = 'solid', size = 1) +
  labs(x='pB', y='L(pB)', title = 'Likelihood') +
  theme_minimal()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```r
print(likelihood_plot)
```



```r
plot(pB_length,df$logLik,type="l",ylab="logLik")
abline(v=pB_length[df$logLik==max(df$logLik)],lty=3,col=2)
```



- The approximated MLE is once again 0.6.
- The plot for the likelihood is much skinnier than the other two, same with the log-likelihood
- $L(P_{B1}) = (1 - P_{B1})^{4000} P_{B1}^{6000}$
  - $logL(P_{B1}) = 4000log(1 - P_{B1}) + 6000log(P_{B1})$
  - $(d/dP_{B1})logL(P_{B1}) = -4000/(1 - P_{B1}) + 6000/P_{B1}$
  - $\rightarrow 0 = -4000/(1 - \hat{P}_{B1}) + 6000/\hat{P}_{B1}$
  - $-6000/\hat{P}_{B1} = -4000/(1 - \hat{P}_{B1})$
  - $-6000 * (1 - \hat{P}_{B1}) = -4000 * \hat{P}_{B1}$
  - $-6000 + 6000 * \hat{P}_{B1} = -4000 * \hat{P}_{B1}$
  - $P_{B1} = 0.6$
- It is incorrect to perform an LRT to compare cases 1, 2, and 3 because you would be comparing the same model, just with more iterations. The models have the same amount of complexity.

Question 2

a)

$$L(\beta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[\frac{-1}{2\sigma^2}(y_i - x_i\beta)\right]$$

$$logL(\beta) = log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[\frac{-1}{2\sigma^2}(y_i - x_i\beta)^2\right]$$

$$= \sum_{i=1}^{n} log \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[\frac{-1}{2\sigma^2}(y_i - x_i\beta)^2\right]$$

$$= \sum_{i=1}^{n} (\left[\frac{-1}{2\sigma^2} (y_i - x_i\beta)^2 \right] - log(\sqrt{2\pi\sigma^2}))$$

$$= \frac{-1}{2\sigma^2} \sum_{i=1}^{n} (y_i - x_i\beta)^2 - nlog(\sqrt{2\pi\sigma^2})$$

b)

$$logL(\beta) = \frac{-1}{2\sigma^2} \sum_{i=1}^{n} (Y - X\beta)^T (Y - X\beta) - nlog(\sqrt{2\pi\sigma^2})$$

$$\frac{d}{d\beta} logL(\beta) = \frac{d}{d\beta} \left\{ \frac{-1}{2\sigma^2} \sum_{i=1}^{n} (Y^T Y - Y^T \beta X - X^T \beta^T Y + \beta^T X^T X\beta) - nlog(\sqrt{2\pi\sigma^2}) \right\}$$

$$= \left\{ \frac{-1}{2\sigma^2} \sum_{i=1}^{n} (0 - 2X^T Y + 2X^T X\beta) - 0 \right\}$$

$$\rightarrow 0 = \left\{ \frac{-1}{2\sigma^2} \sum_{i=1}^{n} (0 - 2X^T Y + 2X^T X\hat{\beta}) \right\}$$

$$0 = \left\{ \frac{-1}{2\sigma^2} (-2X^T Y + 2X^T X\hat{\beta}) \right\}$$

$$0 = -2X^T Y + 2X^T X\hat{\beta}$$

$$X^T Y = X^T X\hat{\beta}$$

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

c)

It is the same as the least squares estimate

d)

$$E(\hat{\beta}) = E((X^T X)^{-1} X^T Y)$$

$$= E((X^T X)^{-1} X^T (X\beta + \epsilon))$$

$$= E((X^T X)^{-1} X^T X\beta + (X^T X)^{-1} X^T \epsilon))$$

$$= E(\beta + (X^T X)^{-1} X^T \epsilon))$$

$$= E(\beta) + E((X^T X)^{-1} X^T \epsilon))$$

$$= \beta + (X^T X)^{-1} X^T E(\epsilon)$$

$$= \beta + (X^T X)^{-1} X^T \times 0$$

$$= \beta$$

The expected value of beta is unbiased.

$$V(\hat{\beta}) = V\left[(X^T X)^{-1} X^T Y\right]$$

$$= \left[(X^T X)^{-1} X^T\right]^2 V(Y)$$

$$= \left[(X^T X)^{-1} X^T\right]^2 \times \sigma^2$$

$$= \left[(X^T X)^{-1} X^T\right]\left[(X^T X)^{-1} X^T\right]^T \times \sigma^2$$

$$= \left[(X^T X)^{-1} X^T\right]\left[X(X^T X)^{-1}\right] \times \sigma^2$$

$$= (X^T X)^{-1} \times \sigma^2$$

The variance of beta is $(X^T X)^{-1} \times \sigma^2$

Question 3

```
suppressMessages(library(boot))
```

```
## Warning: package 'boot' was built under R version 4.3.3
```

```
which(is.na(urine),arr.ind = TRUE)
```

```
##    row col
## 55  55   4
## 1    1   5
```

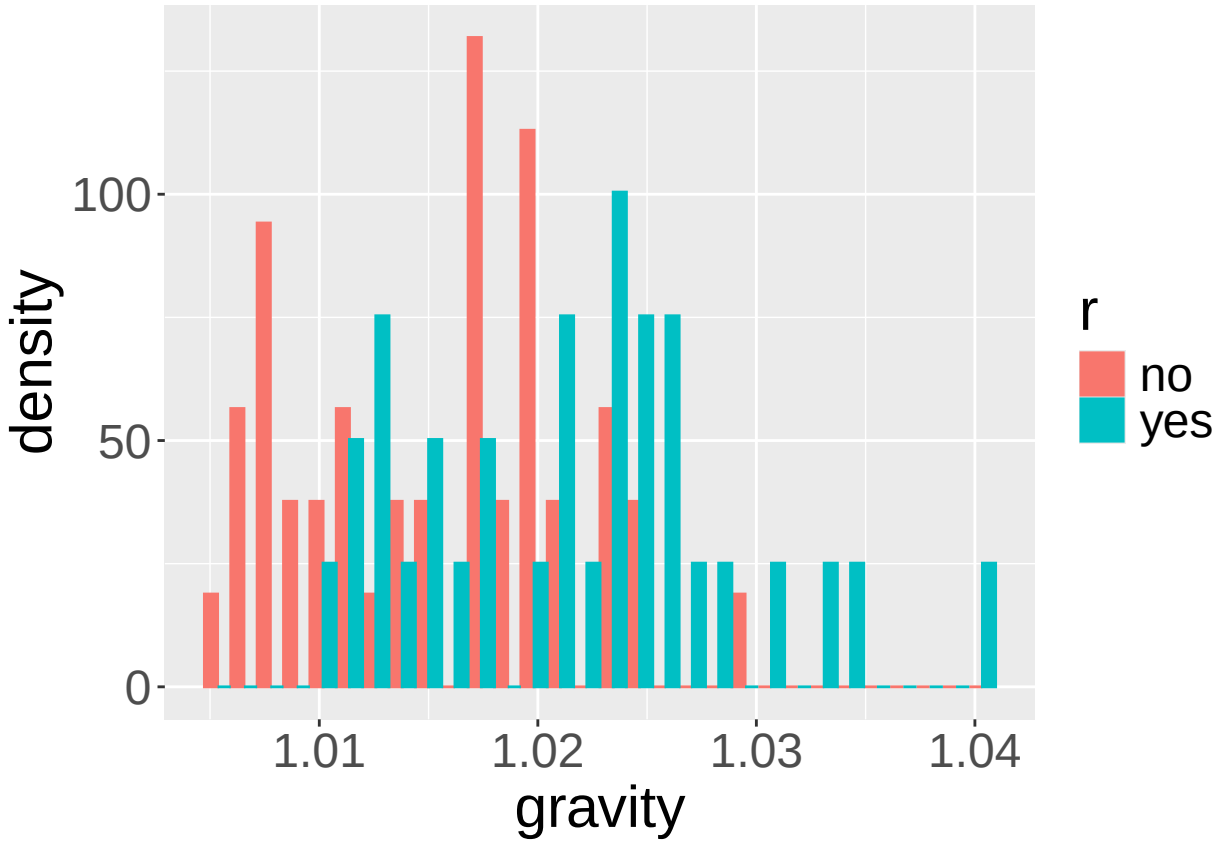
```
urine <- urine[-c(1,55),]
urine$r <- factor(urine$r, levels= c("0","1"),labels = c("no","yes"))
str(urine)
```

```
## 'data.frame':    77 obs. of  7 variables:
##  $ r      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ gravity: num  1.02 1.01 1.01 1 1.02 ...
##  $ ph     : num  5.74 7.2 5.51 6.52 5.27 5.62 5.67 5.41 6.13 6.19 ...
##  $ osmo   : num  577 321 408 187 668 ...
##  $ cond   : num  20 14.9 12.6 7.5 25.3 17.4 35.9 21.9 25.7 11.5 ...
##  $ urea   : num  296 101 224 91 252 195 550 170 382 152 ...
##  $ calc   : num  4.49 2.36 2.15 1.16 3.34 1.4 8.48 1.16 2.21 1.93 ...
```

a)

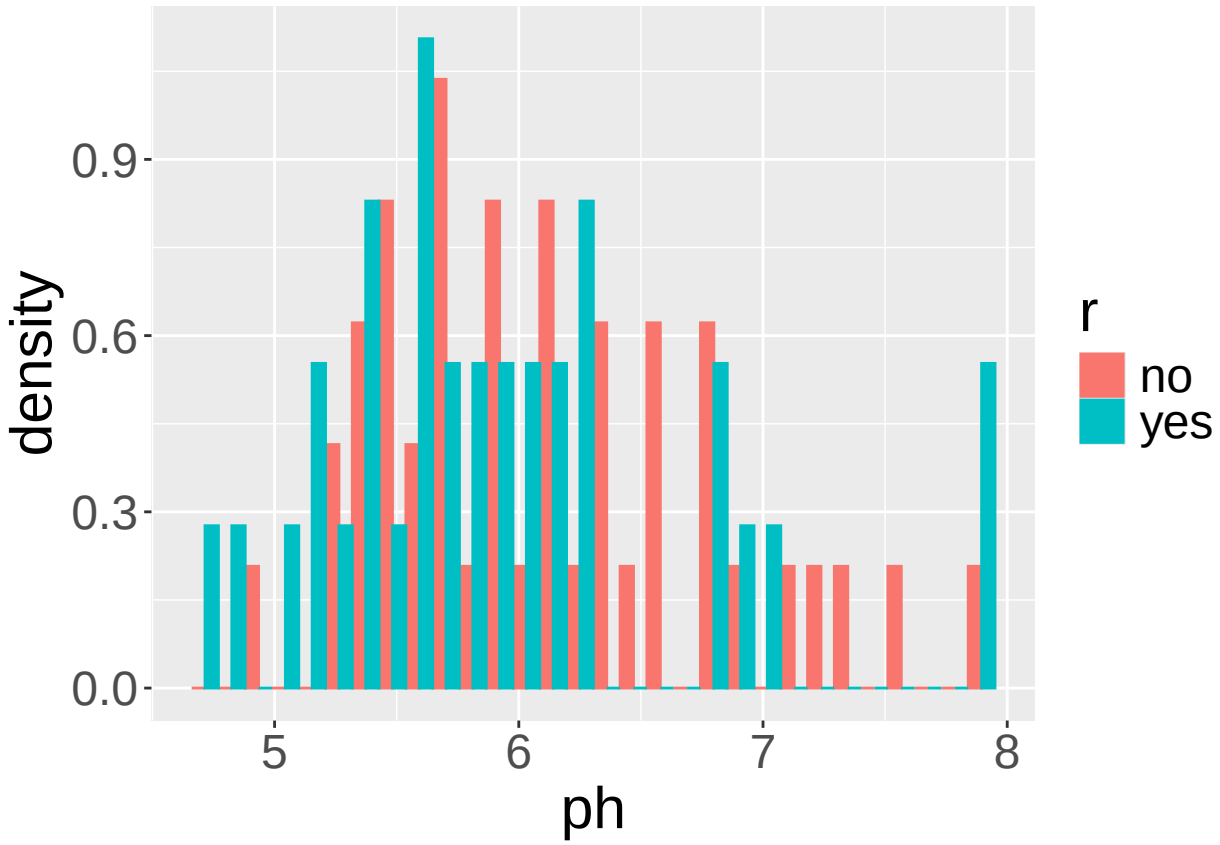
```
suppressMessages(library(ggplot2))

suppressMessages(ggplot(urine, aes(x=gravity, fill=r, color=r)) +
  geom_histogram(position="dodge",  aes(y=after_stat(density))) +
  theme(text=element_text(size=22)))
```

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



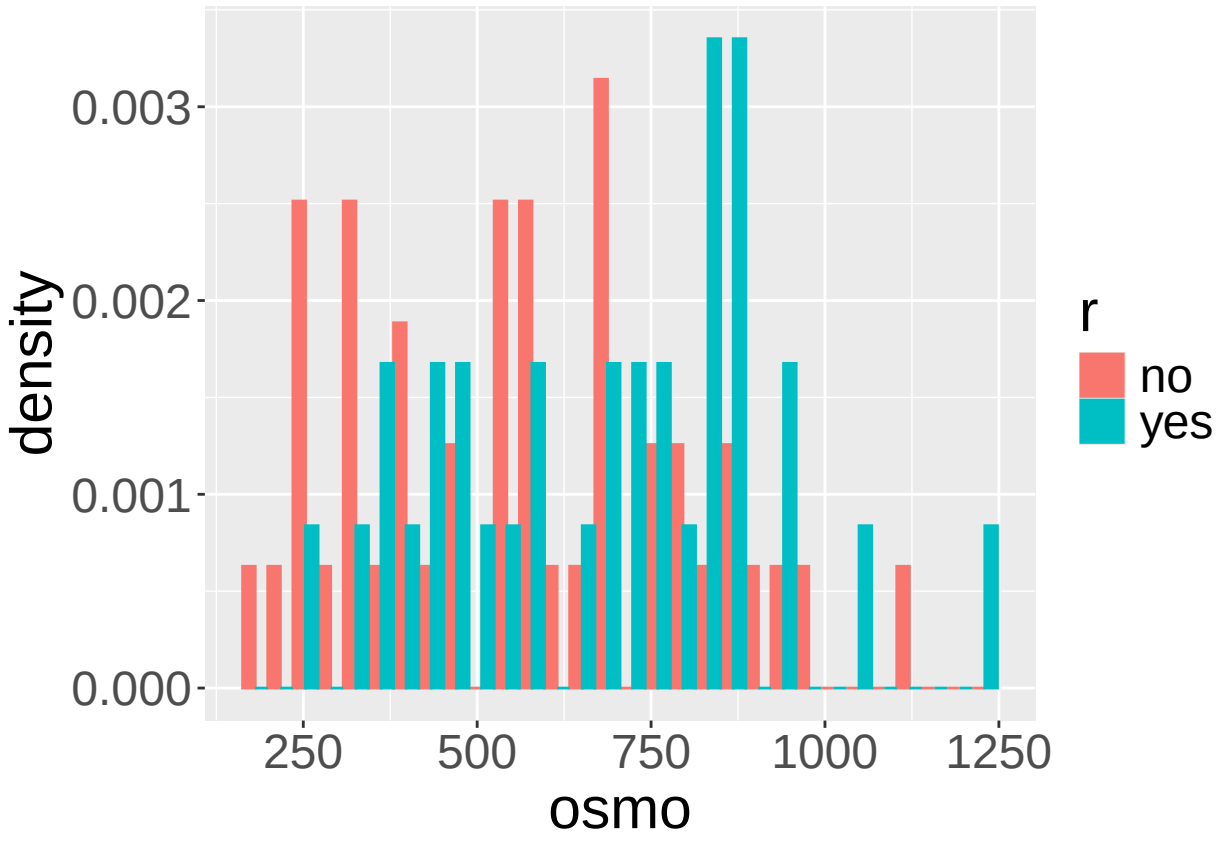

```
suppressMessages(ggplot(urine, aes(x=ph, fill=r, color=r)) +
  geom_histogram(position="dodge", aes(y=after_stat(density))) +
  theme(text=element_text(size=22)))
```

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



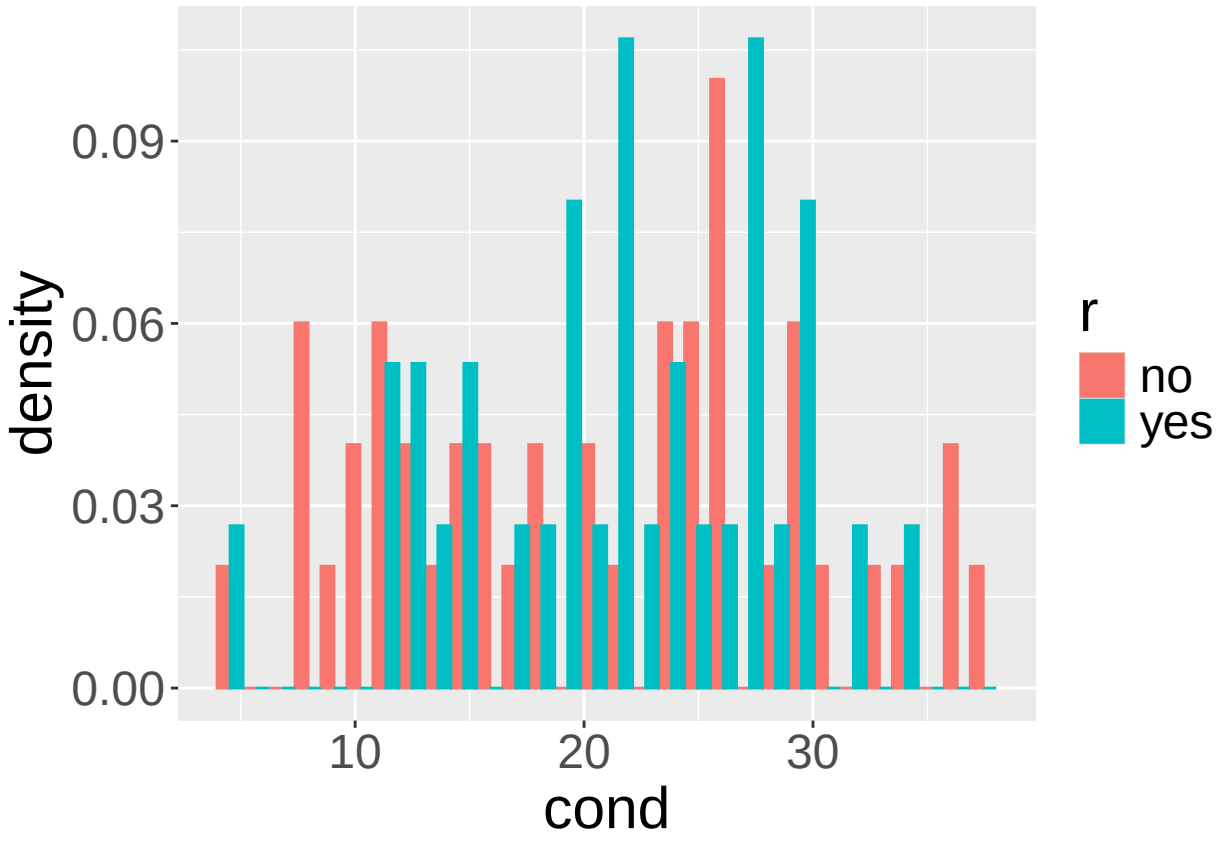

```
suppressMessages(ggplot(urine, aes(x=osmo, fill=r, color=r)) +
  geom_histogram(position="dodge", aes(y=after_stat(density))) +
  theme(text=element_text(size=22)))
```

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



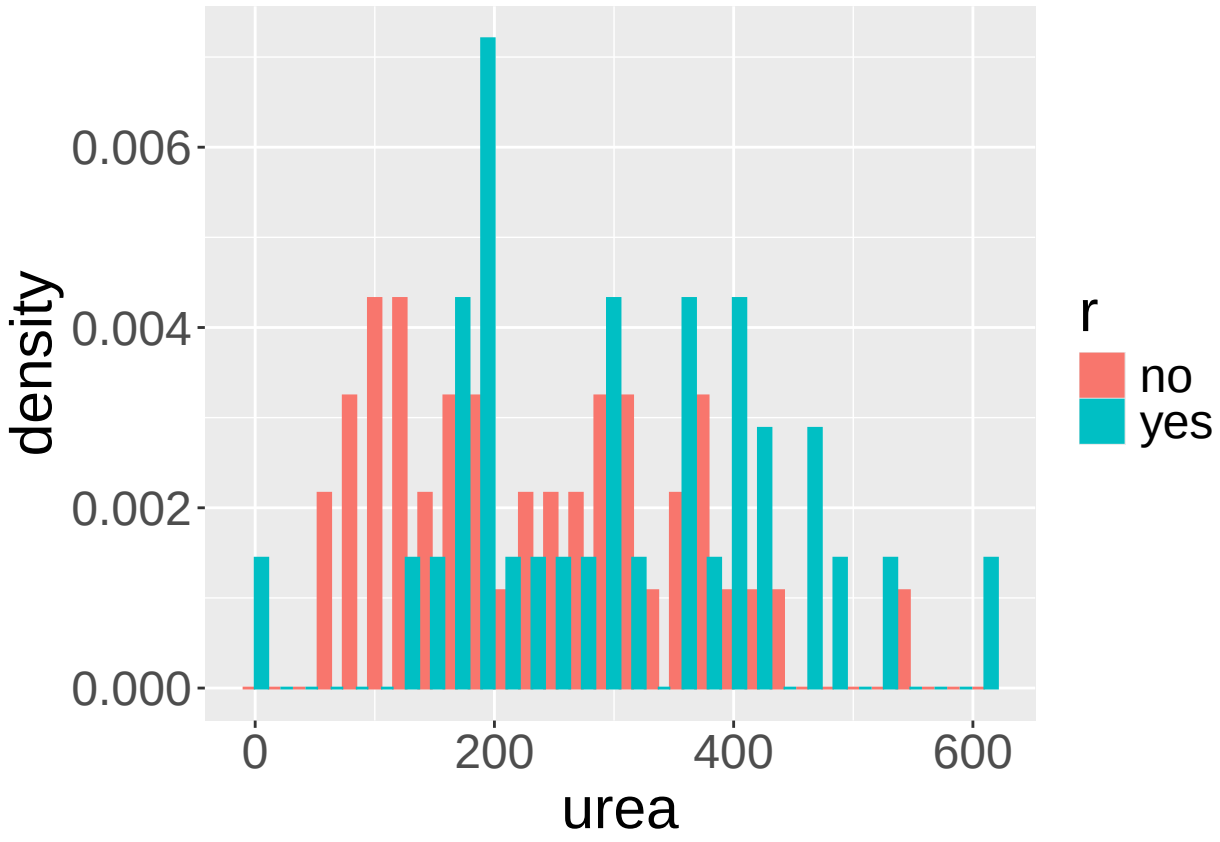

```
suppressMessages(ggplot(urine, aes(x=cond, fill=r, color=r)) +
  geom_histogram(position="dodge", aes(y=after_stat(density))) +
  theme(text=element_text(size=22)))
```

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



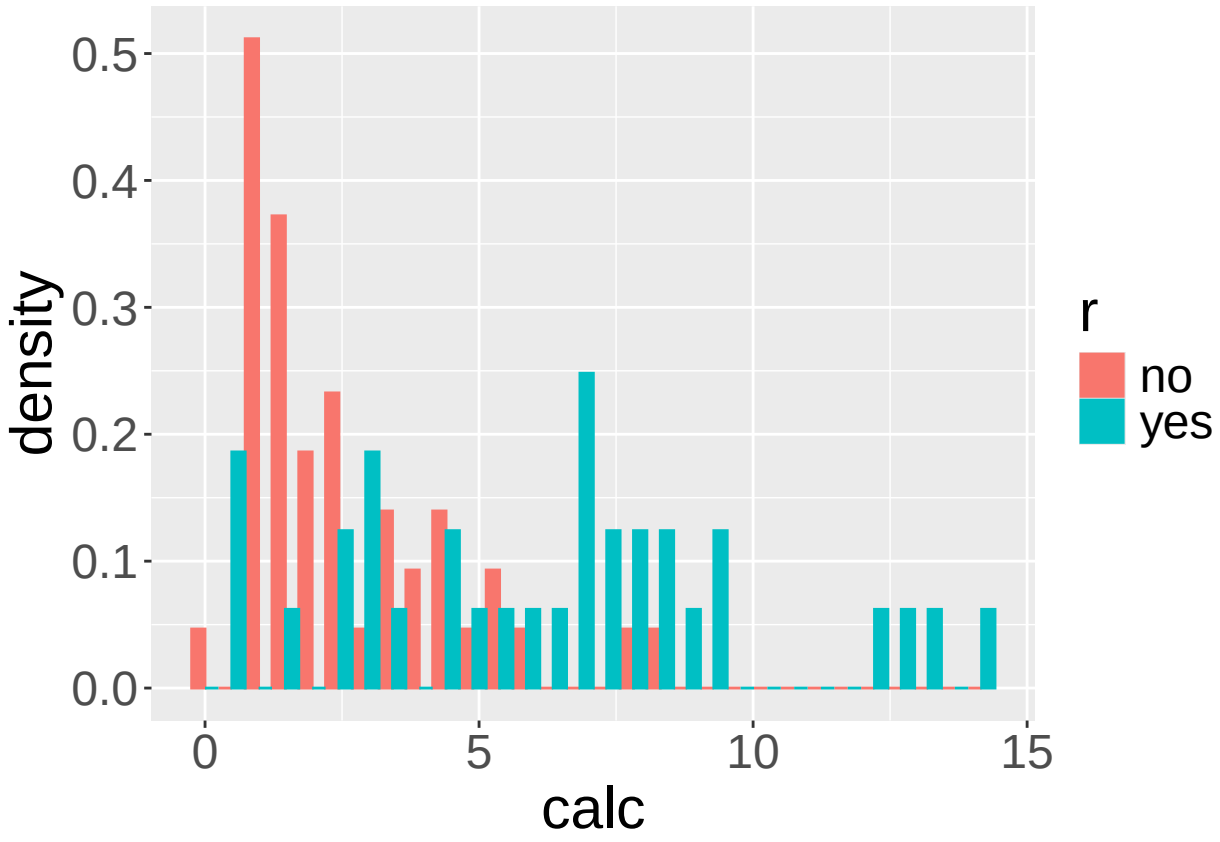

```
ggplot(urine, aes(x=urea, fill=r, color=r)) +
  geom_histogram(position="dodge", aes(y=after_stat(density))) +
  theme(text=element_text(size=22))
```

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




```
ggplot(urine, aes(x=calc, fill=r, color=r)) +
  geom_histogram(position="dodge", aes(y=after_stat(density))) +
  theme(text=element_text(size=22))
```

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



It appears that the calcium concentration, gravity, and osmolarity will be the best covariates in predicting r. This is because they each show a difference in having or not having calcium oxalate crystals depending on the level of the covariate. In calcium concentration, we see more instances of not having crystals when calcium concentration is low. In gravity, we see more instances of not having crystals when gravity is low. Finally, we also see more instances of not having crystals when osmolarity is low.

b)

```
logismodel <- glm(r ~ gravity + ph + osmo + cond + urea + calc, family = binomial, urine)
suppressMessages(library(boot))
summary(logismodel)
```

```
##
## Call:
## glm(formula = r ~ gravity + ph + osmo + cond + urea + calc, family = binomial,
##     data = urine)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -355.33771  222.76696  -1.595  0.11069
## gravity      355.94379  222.11004   1.603  0.10903
## ph            -0.49570    0.56976  -0.870  0.38429
## osmo           0.01681    0.01782   0.944  0.34536
## cond          -0.43282    0.25123  -1.723  0.08493 .
## urea          -0.03201    0.01612  -1.986  0.04703 *
```

```
## calc           0.78369    0.24216   3.236  0.00121 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 105.17  on 76  degrees of freedom
## Residual deviance:  57.56  on 70  degrees of freedom
## AIC: 71.56
##
## Number of Fisher Scoring iterations: 6
```

Residual deviance: 57.56 on 70 degrees of freedom

Deviance cannot be used to assess model fit because it does not follow a chi-squared distribution. Furthermore, the log-likelihood of the saturated model is always 0 in a logistic regression model, so the results would not be useful for our use.

c)

```
interceptlogismodel <- glm(r ~ 1, family = binomial, urine)
anova(logismodel, interceptlogismodel, test='Chi')
```

```
## Analysis of Deviance Table
##
## Model 1: r ~ gravity + ph + osmo + cond + urea + calc
## Model 2: r ~ 1
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        70      57.56
## 2        76     105.17 -6  -47.608 1.415e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis is that the two models are the same. The alternate hypothesis is that the model with more variables is better than the intercept only model. The test statistic is the deviance and it has a Chi-squared distribution. The P-value from the analysis of variance shows we can reject the null hypothesis. It appears that the model with more variables is better than the intercept only model.

d)

```
suppressMessages(library(faraway))
```

```
## Warning: package 'faraway' was built under R version 4.3.3
```

```
## Warning in check_dep_version(): ABI version mismatch:
## lme4 was built with Matrix ABI version 1
## Current Matrix ABI version is 0
## Please re-install lme4 from source or restore original 'Matrix' package
```

```
logistic_fit <- step(logismodel, trace = 0)
sumary(logistic_fit)
```

```
##                Estimate  Std. Error z value Pr(>|z|)
## (Intercept) -5.0001e+02  1.6187e+02 -3.0889 0.002009
## gravity      4.9712e+02  1.6133e+02  3.0814 0.002060
## cond        -2.0547e-01  7.1046e-02 -2.8921 0.003827
## urea        -1.7829e-02  7.2304e-03 -2.4658 0.013672
## calc         7.2232e-01  2.1997e-01  3.2837 0.001025
##
## n = 77 p = 5
## Deviance = 59.07103 Null Deviance = 105.16785 (Difference = 46.09682)
```

The lowest AIC is 69.071 using the combination r~gravity+cond+urea+calc

e)

```
suppressMessages(library(pROC))
```

```
## Warning: package 'pROC' was built under R version 4.3.3
```

```
prediction <- fitted(logistic_fit)
roc_obj <- roc(response=urine$r, predictor=prediction)
```

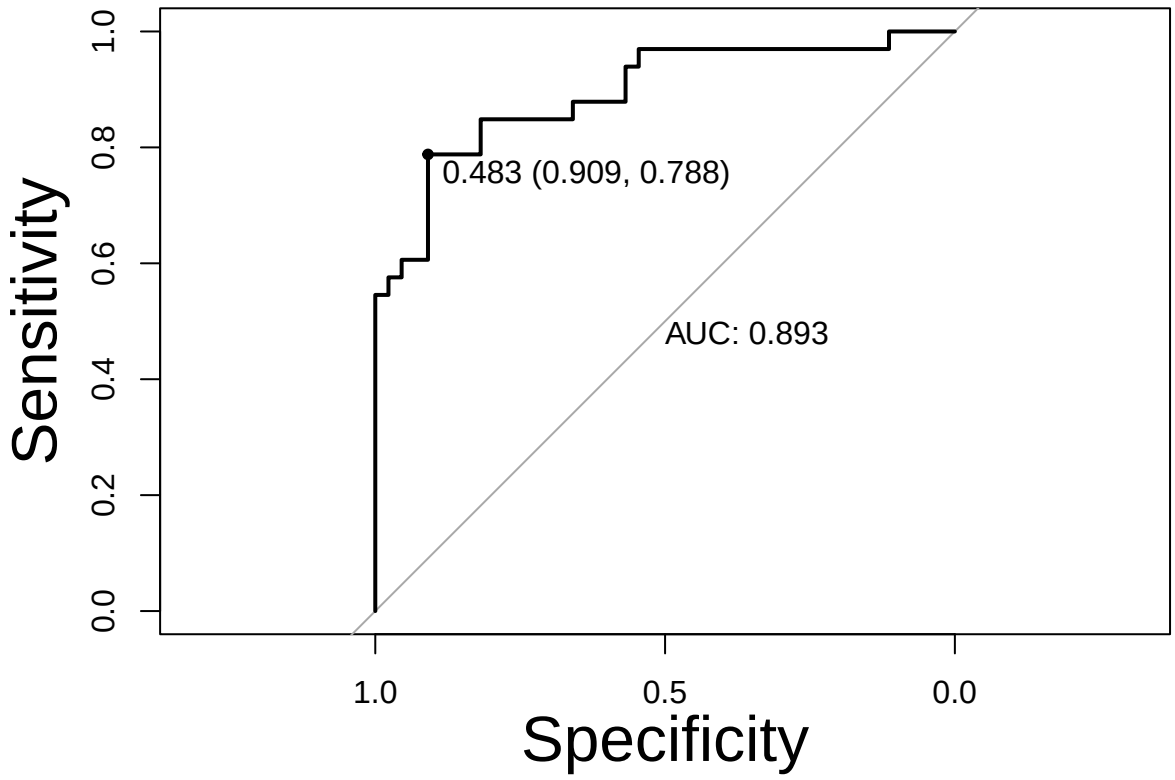
```
## Setting levels: control = no, case = yes
```

```
## Setting direction: controls < cases
```

```
AUC <- auc(roc_obj)
roc_logistic <- c(coords(roc_obj, "b",
ret=c("threshold","se","sp","accuracy"),
best.method="youden"),AUC)
names(roc_logistic) <- c("Threshold","Sensitivity","Specificity",
"Accuracy","AUC")

t(roc_logistic)
```

```
##      Threshold Sensitivity Specificity Accuracy  AUC
## [1,] 0.4830697 0.7878788   0.9090909   0.8571429 0.8932507
```

```
plot(roc_obj,legacy.axes=FALSE,print.auc=TRUE,print.thres=TRUE,cex.lab=2)
```



f)

```
urinedata <- data.frame(urine$r)

suppressMessages(library(dplyr))
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
dataurineext <- mutate(urinedata, predprob=prediction)

dataurineext <- mutate(dataurineext, predout=ifelse(predprob < 0.483, "no", "yes"))
xtabs( ~ urine$r + predout, dataurineext)
```

```
##        predout
## urine$r no yes
##     no  40   4
##     yes  7  26
```

- True positive rate: $26/(26+7) = 0.787878$
- False positive rate: $4/(4+40) = 0.09090909$
- Positive predictive value: $26/(4+26) = 0.86666$
- Negative predictive value: $40/(40+7) = 0.8510638$

g)

The fitted logistic model has an impressive specificity rate, but could have a better sensitivity. The area under curve statistic is quite high and depicts that the model is fit well to the response. The false positive rate is low, which is good to see as well. Overall, the fitted logistic model is generally well-fit to predict the presence of oxilate crystals in urine.

h)

```
set.seed(10)
train_ind <- sample(1:77,51)
train_urine <- urine[train_ind,]
test_urine <- urine[-train_ind,]
```

1)

```
logismodel2 <- glm(r ~ gravity + ph + osmo + cond + urea + calc, family = binomial, train_urine)
logistic_fit2 <- step(logismodel2, trace = 0)
sumary(logistic_fit2)
```

```
##                Estimate  Std. Error z value Pr(>|z|)
## (Intercept) -716.061220  273.920175 -2.6141 0.008946
## gravity      711.896895  272.654711  2.6110 0.009028
## cond          -0.322226    0.116878 -2.7570 0.005834
## urea          -0.021363    0.011014 -1.9397 0.052415
## calc           0.937711    0.361683  2.5926 0.009524
##
## n = 51 p = 5
## Deviance = 31.76421 Null Deviance = 70.52444 (Difference = 38.76023)
```

The same covariates remain in the model

2)

```
suppressMessages(library(pROC))
prediction2 <- predict(logistic_fit2, newdata = test_urine, type = "response")
roc_obj2 <- roc(response=test_urine$r, predictor=prediction2)
```

```
## Setting levels: control = no, case = yes
```

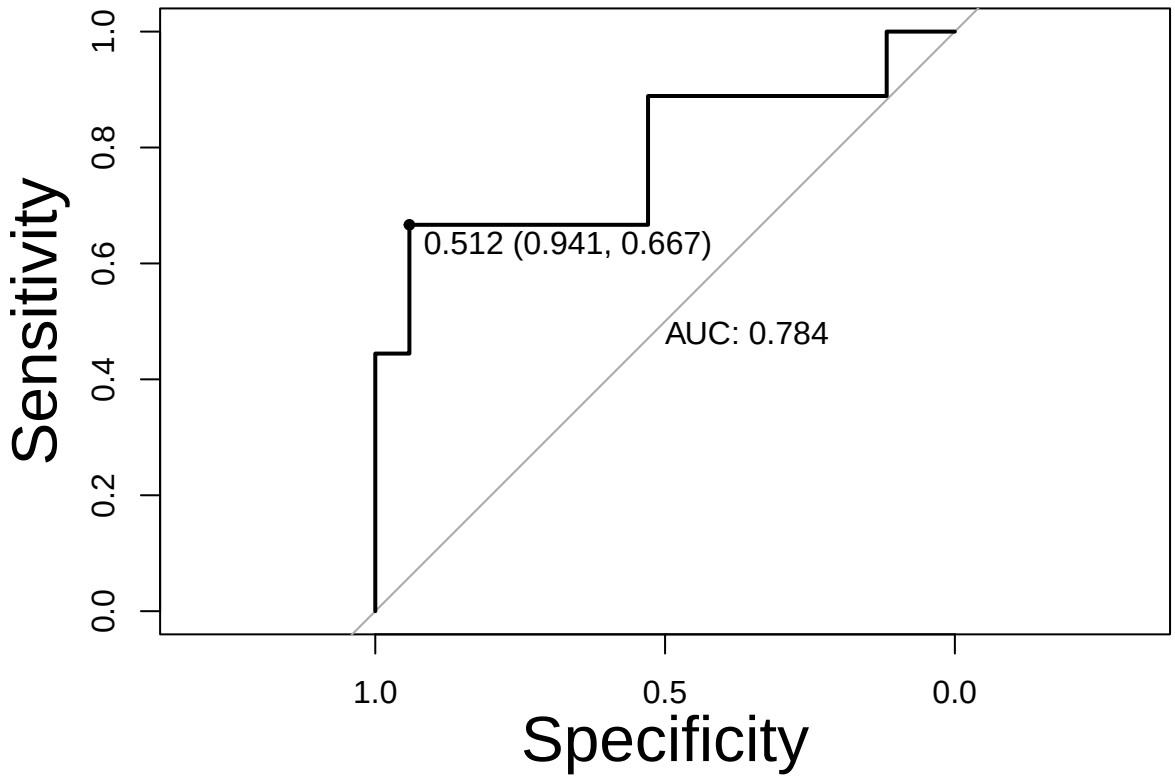
```
## Setting direction: controls < cases
```

```
AUC2 <- auc(roc_obj2)
roc_logistic2 <- c(coords(roc_obj2, "b",
ret=c("threshold","se","sp","accuracy"),
best.method="youden"),AUC2)
```

```
names(roc_logistic) <- c("Threshold","Sensitivity","Specificity",
"Accuracy","AUC")

t(roc_logistic2)
```

```
##      threshold sensitivity specificity accuracy
## [1,] 0.5115104 0.6666667   0.9411765   0.8461538 0.7843137
```

```
plot(roc_obj2,legacy.axes=FALSE,print.auc=TRUE,print.thres=TRUE,cex.lab=2)
```



Sensitivity got worse and specificity got better. AUC got worse. Overall, should be worse than the first model

```
urinedata2 <- data.frame(test_urine$r)

suppressMessages(library(dplyr))
dataurineext2 <- mutate(urinedata2, predprob=prediction2)

dataurineext2 <- mutate(dataurineext2, predout=ifelse(predprob < 0.512, "no", "yes"))
xtabs( ~ test_urine$r + predout, dataurineext2)
```

```
##             predout
## test_urine$r no yes
##          no  16   1
##          yes  3   6
```

- True positive rate: $6/(6+3) = 0.6666667$

- False positive rate: 1/(1+16) = 0.05882353
- Positive predictive value: 6/(1+6) = 0.8571429
- Negative predictive value: 16/(16+3) = 0.8421053

The model has the same variables and performs similarly with the test data as it did with the training data. Our model has good predictive strength.

Question 4

```
suppressMessages(library(faraway))
data(seeds)
## creating a new predictor describing the box:
seeds$box <- factor(x=rep(1:8, c(6,6,6,6,6,6,6,6)), levels=c("1","2","3","4","5","6","7","8"))
## removing one observation with missing data
(seeds[is.na(seeds$germ),])
```

```
##    germ moisture covered box
## 47   NA        9     yes   8
```

```
seeds <- seeds[!is.na(seeds$germ),]
str(seeds)
```

```
## 'data.frame':    47 obs. of  4 variables:
##  $ germ    : num  22 41 66 82 79 0 25 46 72 73 ...
##  $ moisture: num  1 3 5 7 9 11 1 3 5 7 ...
##  $ covered : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ box     : Factor w/ 8 levels "1","2","3","4",..: 1 1 1 1 1 1 2 2 2 2 ...
```

a)

```
binmod <- glm(cbind(germ, 100-germ) ~ moisture + box, family = binomial, data = seeds)
sumary(binmod)
```

```
##               Estimate Std. Error  z value  Pr(>|z|)
## (Intercept)  0.5938719  0.0983187   6.0403 1.539e-09
## moisture    -0.1104873  0.0088127 -12.5373 < 2.2e-16
## box2        -0.0414929  0.1176091  -0.3528    0.7242
## box3        -0.0414929  0.1176091  -0.3528    0.7242
## box4         0.0207244  0.1175436   0.1763    0.8600
## box5        -0.0900120  0.1176965  -0.7648    0.4444
## box6        -0.0622695  0.1176427  -0.5293    0.5966
## box7        -0.0692004  0.1176552  -0.5882    0.5564
## box8         0.0672974  0.1234101   0.5453    0.5855
##
## n = 47 p = 9
## Deviance = 1624.40423 Null Deviance = 1790.99166 (Difference = 166.58743)
```

b)

The coefficient for moisture is the log-odds decrease in the odds ratio of the germination of 100 seeds per 1 unit increase of moisture. In this case, $e^{-0.1104873} = 0.8954$, which translates to a 10.46% decrease in the odds of germination. The coefficient for box4 is 0.020724, which means that putting the seeds in box 4 would increase the odds of seeds germinated per 100 by $e^{-0.1104873} = 1.02094$, which is 2.094%.

c)

We can use the z-value test and the deviance-based Hosmer-Lemeshow test.

```
library(ResourceSelection)
```

```
## Warning: package 'ResourceSelection' was built under R version 4.3.3
```

```
## ResourceSelection 0.3-6   2023-06-27
```

```
hoslem.test(seeds$germ,fitted(binmod), g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  seeds$germ, fitted(binmod)
## X-squared = 547044, df = 8, p-value < 2.2e-16
```

Since the p-value is less than 0.05, we reject the null hypothesis. This means that the model is not fit well.

d)

- Wrong form of the model: missing relevant predictors or transformations
- Presence of outliers
- Sparse data

e)

```
(sigma2 <- sum(residuals(binmod,type="pearson")**2)/(47-9))
```

```
## [1] 35.71223
```

This is over-dispersion. The variance is larger than expected for a dataset of this type (germinated seeds out of 100)

f)

```r
binmod0 <- glm(cbind(germ, 100-germ) ~ moisture, family = binomial, data = seeds)

F1 <- ((deviance(binmod0)-deviance(binmod))/
(df.residual(binmod0)-df.residual(binmod)))/sigma2

binmod01 <- glm(cbind(germ, 100-germ) ~ box, family = binomial, data = seeds)

F2 <- ((deviance(binmod01)-deviance(binmod))/
(df.residual(binmod01)-df.residual(binmod)))/sigma2
```

For the first separate model,

- The F-statistic accounting for dispersion for moisture is 0.010281825176862
- The critical value at 0.05 is F(8), 38 = 2.27
- we do not reject the null hypothesis that the smaller model is just as good as the larger model, the moisture variable explains the response well

For the second separate model,

- The F-statistic accounting for dispersion for box is 4.53882775737736
- The critical value at 0.05 is F(1), 38 = 4.17
- we reject the null hypothesis that the smaller model is just as good as the larger model, the box variable does not explain the response well

g)

```r
anova(binmod, binmod0, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(germ, 100 - germ) ~ moisture + box
## Model 2: cbind(germ, 100 - germ) ~ moisture
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        38     1624.4
## 2        45     1627.0 -7  -2.5703   0.9217
```

```r
anova(binmod, binmod01, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(germ, 100 - germ) ~ moisture + box
## Model 2: cbind(germ, 100 - germ) ~ box
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        38     1624.4
## 2        39     1786.5 -1  -162.09 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I believe the question is asking how the results differ from **f)**, not **e)**. If this is the case, the difference between these two tests is that when you ignore over-dispersion, the *box* variable is very clearly not helping the model's fit by looking at the p-value between models. When you account for over-dispersion, the test statistic is barely greater than the critical value.

Question 5

a)

$$L(\lambda) = \prod_{i=1}^{n} \frac{\lambda^{y_i} e^{-\lambda}}{y_i!}$$

$$LogL(\lambda) = \sum_{i=1}^{n} (y_i log(\lambda) - \lambda - log(y_i!))$$

$$= -n\lambda + log(\lambda) \sum_{i=1}^{n} y_i - \sum_{i=1}^{n} log(y_i!)$$

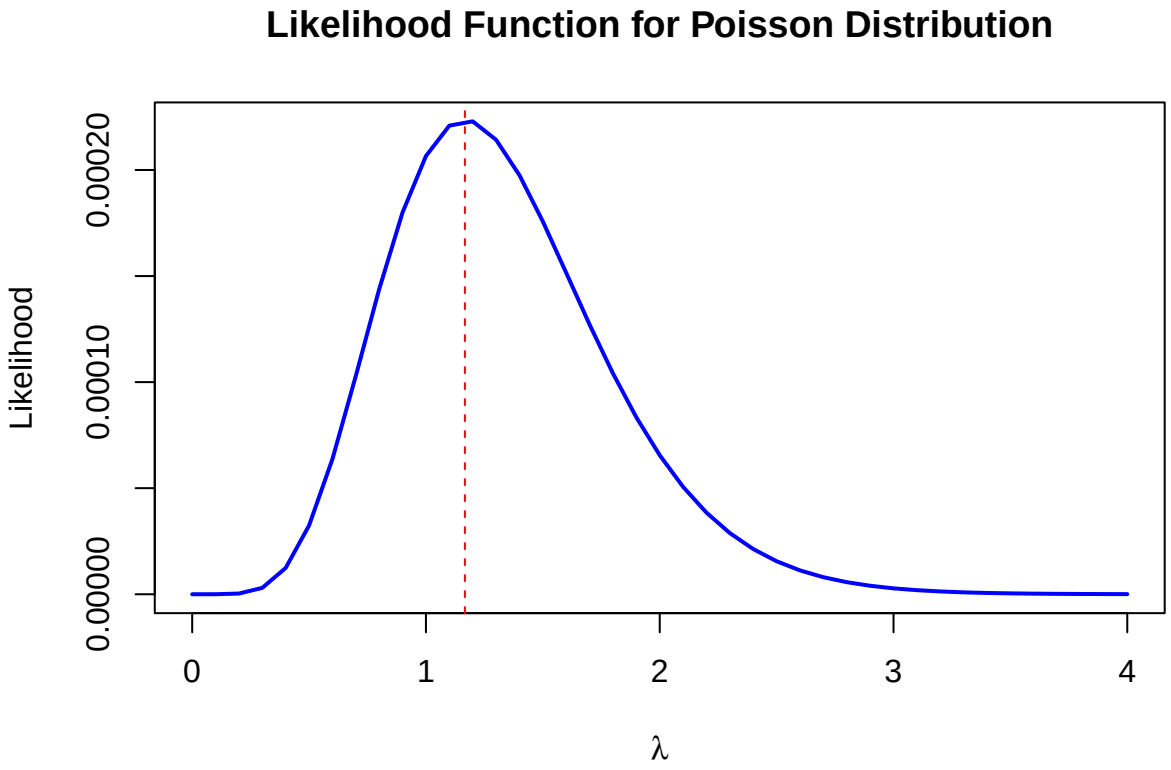
LogL(λ), with all the data fit, is $[-6\lambda + log(\lambda) * 7 - log(12)]$

b)

Intuitively, λ would be just over 1, since the derivative of the log-likelihood would give you a value near 7/6, since the derivative with respect to λ gets rid of the log(12) term.

c)

```
location <- factor(c(0, 1, 1, 1, 0, 0), levels = c(0,1), labels = c("home", "work"))
cigarettes <- c(3, 0, 0, 1, 2, 1)

poisson_likelihood <- function(lambda, data) {
  prod(dpois(data,lambda))
}

poisson_neglikelihood <- function(lambda, data) {
  -prod(dpois(data,lambda))
}

lambda_values <- seq(0, 4, by = 0.1)
likelihood_values <- sapply(lambda_values, poisson_likelihood, data = cigarettes)

plot(lambda_values, likelihood_values, type = "l",
     col = "blue", lwd = 2,
     xlab = expression(lambda),
     ylab = "Likelihood",
     main = "Likelihood Function for Poisson Distribution")

mle <- mean(cigarettes)
abline(v = mle, col = "red", lty = 2)
```




```
result <- optim(
  par = 1,
  fn = poisson_neglikelihood,
  data = cigarettes,
  method = "L-BFGS-B", # Bounded optimization
  lower = 0.001        # Lambda must be > 0
)

result$par
```

```
## [1] 1.166666
```

d)

Work = 0, Home = 1

$$LogL(\lambda_W, \lambda_H) = \sum_{i=1}^{n}(y_i log(\lambda_W) - \lambda_W - log(y_i!)) + \sum_{j=1}^{m}(y_j log(\lambda_H) - \lambda_H - log(y_j!))$$

$$= -n\lambda_W + log(\lambda_W)\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} log(y_i!) + (-m\lambda_H + log(\lambda_H)\sum_{j=1}^{m} y_j - \sum_{j=1}^{m} log(y_j!))$$

With all data fit, $LogL(\lambda_H, \lambda_W) = -3\lambda_H + log(\lambda_H) * 6 - log(12) + (-3\lambda_W + log(\lambda_W) * 1 - 0))$

e)

Intuitively, the value for λ_H should be around 2, while the value for λ_W should be near 0.33

f)

```
resultwork <- optim(
  par = 1,
  fn = poisson_neglikelihood,
  data = c(0,0,1),
  method = "L-BFGS-B",
  lower = 0.001        # Lambda must be > 0
)

resultwork$par
```

```
## [1] 0.3333343
```

```
resulthome <- optim(
  par = 1,
  fn = poisson_neglikelihood,
  data = c(3,2,1),
  method = "L-BFGS-B",
  lower = 0.001        # Lambda must be > 0
)

resulthome$par
```

```
## [1] 2
```

g)

$$LogL(\lambda) = \sum_{i=1}^{n}(y_i log(\lambda) - \lambda - log(y_i!))$$

$$logL(\beta_0, \beta_1) = \sum_{i=1}^{n}(\beta_0 + \beta_1 x_i)y_i - \sum_{i=1}^{n} log(y_i!) - \sum_{i=1}^{n} e^{\beta_0 + \beta_1 x_i}$$

$$logL(\beta_0, \beta_1) = n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} log(y_i!) - e^{\beta_0} \sum_{i=1}^{n} e^{\beta_1 x_i}$$

$$= n\beta_0 + \beta_1 \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} log(y_i!) - e^{\beta_0} \sum_{i=1}^{n} e^{\beta_1 x_i}$$

$$= n\beta_0 + \beta_1(1) - log(12) - e^{\beta_0}(3e^{\beta_1} + 3)$$

h)

```r
location <- c(0, 1, 1, 1, 0, 0)
cigarettes <- c(3, 0, 0, 1, 2, 1)

poisson_negloglikelihood_betas <- function(params, location, cigarettes) {
  beta0 = params[1]
  beta1 = params[2]
  lambda = exp(beta0 + beta1 * location)
  -sum(cigarettes*log(lambda) - lambda - log(factorial(cigarettes)))
}

resultbetas <- optim(
  par = c(0,0),
  fn = poisson_negloglikelihood_betas,
  location = location,
  cigarettes = cigarettes,
  method = "BFGS"
)

resultbetas$par # betas
```

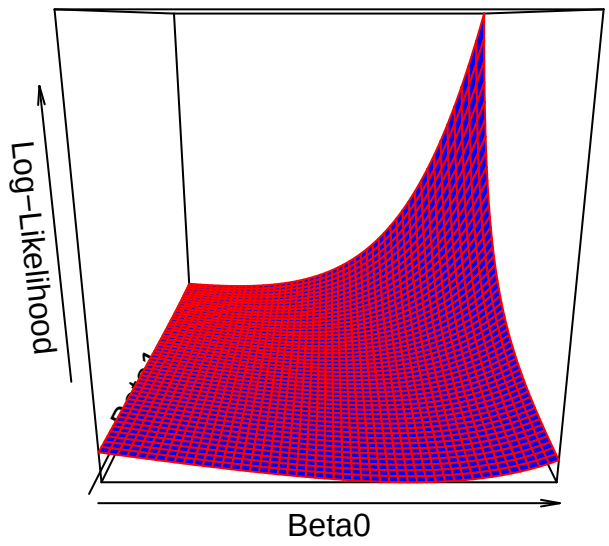
```
## [1]  0.6931484 -1.7917671
```

3D plot of log-likelihood function

```r
library(rgl)
```

```
## Warning: package 'rgl' was built under R version 4.3.3
```

```r
beta0_values <- seq(-2, 2, length.out = 50)
beta1_values <- seq(-2, 2, length.out = 50)

loglik_matrix <- matrix(NA, nrow = length(beta0_values), ncol = length(beta1_values))

for (i in 1:length(beta0_values)) {
  for (j in 1:length(beta1_values)) {
    loglik_matrix[i, j] <- poisson_negloglikelihood_betas(c(
      beta0_values[i], beta1_values[j]), location, cigarettes)
  }
}

persp(beta0_values, beta1_values, loglik_matrix,
      xlab = "Beta0", ylab = "Beta1", zlab = "Log-Likelihood",
      col = "blue", border = "red", main = "3D Log-Likelihood Surface")
```



i)

```
location <- factor(c(0, 1, 1, 1, 0, 0), levels = c(0,1), labels = c("work", "home"))
poisson_cigs_m3 <- glm(cigarettes ~ location, family = poisson)
summary(poisson_cigs_m3) # agrees with model 3
```

```
##
## Call:
## glm(formula = cigarettes ~ location, family = poisson)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.6931     0.4082   1.698   0.0895 .
## locationhome  -1.7918     1.0801  -1.659   0.0971 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 7.2062  on 5  degrees of freedom
## Residual deviance: 3.2437  on 4  degrees of freedom
## AIC: 16.849
##
## Number of Fisher Scoring iterations: 5
```

```
poisson_cigs_m1 <- glm(cigarettes ~ 1, family = poisson)
lambda <- exp(coef(poisson_cigs_m1))
lambda # agrees with model 1
```

```
## (Intercept)
##    1.166667
```

To go from model 3 to model 2, we need to put the coefficients from model 3 into an exponential function. For λ_W, bringing β_0 up to e^{β_0} gives us $e^{0.6931} = 1.99906 \approx 2 = MLE$ from model 2. For λ_H, bringing β_0 and β_1 up to $e^{\beta_0+\beta_1}$ gives us $e^{0.6931-1.7918} = 0.3333041 \approx 0.3333343 = MLE$ from model 2.