MIDTERM_STATCOMP_2020

Data

Reading the data

```
DATA=read.table('https://raw.githubusercontent.com/gdlc/STAT_COMP/master/crab.txt',header=TRUE)
```

Transforming spine and color into factors

[Some of missed this important step]

```
# Formatting spine and colors to factor
 DATA$spine=as.factor(DATA$spine)
 DATA$color=as.factor(DATA$color)
 str(DATA)
```

```
## 'data.frame':    173 obs. of  5 variables:
##  $ color      : Factor w/ 4 levels "2","3","4","5": 2 3 1 3 3 2 1 3 2 3 ...
##  $ spine      : Factor w/ 3 levels "1","2","3": 3 3 1 3 3 3 1 2 1 3 ...
##  $ width      : num  28.3 22.5 26 24.8 26 23.8 26.5 24.7 23.7 25.6 ...
##  $ nSatellites: int  8 0 9 0 4 0 0 0 0 0 ...
##  $ weight     : int  3050 1550 2300 2100 2600 2100 2350 1900 1950 2150 ...
```

Question 1

Fitting the linear model

[Note: since we transformed color and spine to factors, the linear model includes for each of them as many dummy variables as number of levels minus one.]

```
fmLM=lm(nSatellites~color+spine+width +weight ,data=DATA)
summary(fmLM)
```

```
##
## Call:
## lm(formula = nSatellites ~ color + spine + width + weight, data = DATA)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5531 -2.1035 -0.6611  1.4527 11.1435
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.0807123  4.7198950  -0.229   0.8192
## color3      -0.6970915  0.9656844  -0.722   0.4714
## color4      -1.3325025  1.0673232  -1.248   0.2136
## color5      -1.3085385  1.1758897  -1.113   0.2674
## spine2      -0.4526120  0.9522546  -0.475   0.6352
## spine3       0.0646673  0.6250446   0.103   0.9177
## width        0.0230501  0.2392077   0.096   0.9234
```

```
## weight       0.0017544  0.0008579   2.045   0.0424 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.962 on 165 degrees of freedom
## Multiple R-squared:  0.1511, Adjusted R-squared:  0.1151
## F-statistic: 4.195 on 7 and 165 DF,  p-value: 0.000279
```

Conclusions

None of the effects is clearly significant in the linear model, except weight, which has a marginally significant effect.

Question 2

2.1 Poisson Regression

[Note: some of you did not specify the correct family and link.]

```r
fmGLM=glm(nSatellites~color+spine+width +weight,family=poisson(link=log) ,data=DATA)
summary(fmGLM)
```

```
##
## Call:
## glm(formula = nSatellites ~ color + spine + width + weight, family = poisson(link = log),
##     data = DATA)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0290  -1.8630  -0.5988   0.9331   4.9446
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.3618003  0.9665506  -0.374  0.70817
## color3      -0.2648512  0.1681107  -1.575  0.11515
## color4      -0.5137051  0.1953624  -2.629  0.00855 **
## color5      -0.5308601  0.2269157  -2.339  0.01931 *
## spine2      -0.1503718  0.2135754  -0.704  0.48139
## spine3       0.0872826  0.1199287   0.728  0.46674
## width        0.0167487  0.0489197   0.342  0.73207
## weight       0.0004965  0.0001663   2.986  0.00283 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 549.59  on 165  degrees of freedom
## AIC: 920.88
##
## Number of Fisher Scoring iterations: 6
```

Conclusions

When we use a Poisson regression weight is highly significant, and color as a factor also appears significant, with colors 4 and 5 having less satellites than the reference color (1).

[Note: if we want to test whether at least one level of a factor has an effect different than zero we should test the model we fitted against a model that does not include that factor.]

2.2 Apprximate Confidence Intervals

In large samples, Maximum Likelihood Estimates, follow normal distributions. Therefore we can construct 95% CIs using `estimate +/- 1.96*SE`. Only the CIs for color 4 and 5, and for weight, do not contain zero.

```
EST=coef(fmGLM)
SE=summary(fmGLM)$coef[,2] # alternatively sqrt(diag(vcov(fmGLM)))
CI=cbind('Estimate'=EST,'Low'=EST-1.96*SE , 'Up'=EST+1.96*SE)
round(CI,6)
```

```
##              Estimate       Low        Up
## (Intercept) -0.361800 -2.256239  1.532639
## color3      -0.264851 -0.594348  0.064646
## color4      -0.513705 -0.896615 -0.130795
## color5      -0.530860 -0.975615 -0.086105
## spine2      -0.150372 -0.568980  0.268236
## spine3       0.087283 -0.147778  0.322343
## width        0.016749 -0.079134  0.112631
## weight       0.000496  0.000171  0.000822
```

Note: above I used -1.96 and 1.96 in place of `qnorm(p=.025)` and `qnorm(p=.975)`, respectively, if you want to be more precise, replace above -/+1.96 by the more precise quantile.

```
EST=coef(fmGLM)
SE=summary(fmGLM)$coef[,2] # alternatively sqrt(diag(vcov(fmGLM)))
CI=cbind('Estimate'=EST,'Low'=EST+qnorm(p=0.025)*SE , 'Up'=EST+qnorm(p=0.975)*SE)
round(CI,6)
```

```
##              Estimate       Low        Up
## (Intercept) -0.361800 -2.256205  1.532604
## color3      -0.264851 -0.594342  0.064640
## color4      -0.513705 -0.896608 -0.130802
## color5      -0.530860 -0.975607 -0.086113
## spine2      -0.150372 -0.568972  0.268228
## spine3       0.087283 -0.147773  0.322339
## width        0.016749 -0.079132  0.112629
## weight       0.000496  0.000171  0.000822
```

Finally, if you want to be more precise, since we are estimating the error variance, we can use a t-distribution. The DF is sample size minus number of parameters in the model. These CIs are a bit wider because they account for the uncertainty about the error variance.

```
EST=coef(fmGLM)
SE=summary(fmGLM)$coef[,2] # alternatively sqrt(diag(vcov(fmGLM)))
DF=nrow(DATA)-length(EST)

CI=cbind('Estimate'=EST,'Low'=EST+qt(df=DF,p=0.025)*SE , 'Up'=EST+qt(df=DF,p=0.975)*SE)
round(CI,6)
```

```
##              Estimate       Low        Up
## (Intercept) -0.361800 -2.270202  1.546601
## color3      -0.264851 -0.596777  0.067074
## color4      -0.513705 -0.899437 -0.127973
## color5      -0.530860 -0.978893 -0.082827
## spine2      -0.150372 -0.572065  0.271321
```

```
## spine3       0.087283 -0.149510  0.324075
## width        0.016749 -0.079841  0.113338
## weight       0.000496  0.000168  0.000825
```

Question 3

```
B=5000
EST=matrix(nrow=B,ncol=length(coef(fmGLM)),NA)
colnames(EST)=names(coef(fmGLM))

for(i in 1:B){

  tmp=sample(1:nrow(DATA),size=nrow(DATA),replace=TRUE)
  fm=glm(nSatellites~color+spine+width +weight,family=poisson(link=log) ,data=DATA[tmp,])
  EST[i,]=coef(fm)
}

 SE=apply(FUN=sd,X=EST,MARGIN=2,na.rm=TRUE)
 CI=apply(FUN=quantile,prob=c(.025,.975),X=EST,MARGIN=2,na.rm=TRUE)
 print(round(SE,6))
```

```
## (Intercept)      color3      color4      color5      spine2      spine3
##    1.530894    0.401726    0.429256    0.775712    0.603423    0.245805
##       width      weight
##    0.076373    0.000286
```

```
 message('Estimate and Boostrap SE and 95%CI')
```

```
## Estimate and Boostrap SE and 95%CI
```

```
 print(round(cbind('Estimate'=coef(fmGLM),'SE'=SE,CI=t(CI)),6))
```

```
##              Estimate       SE      2.5%    97.5%
## (Intercept) -0.361800 1.530894 -3.405030 2.639391
## color3      -0.264851 0.401726 -0.859362 0.425032
## color4      -0.513705 0.429256 -1.210793 0.226243
## color5      -0.530860 0.775712 -1.711644 0.364254
## spine2      -0.150372 0.603423 -1.026390 0.510224
## spine3       0.087283 0.245805 -0.362198 0.613491
## width        0.016749 0.076373 -0.138820 0.160413
## weight       0.000496 0.000286  0.000021 0.001158
```

```
 message('For comparsion, here are the asymptotic SEs')
```

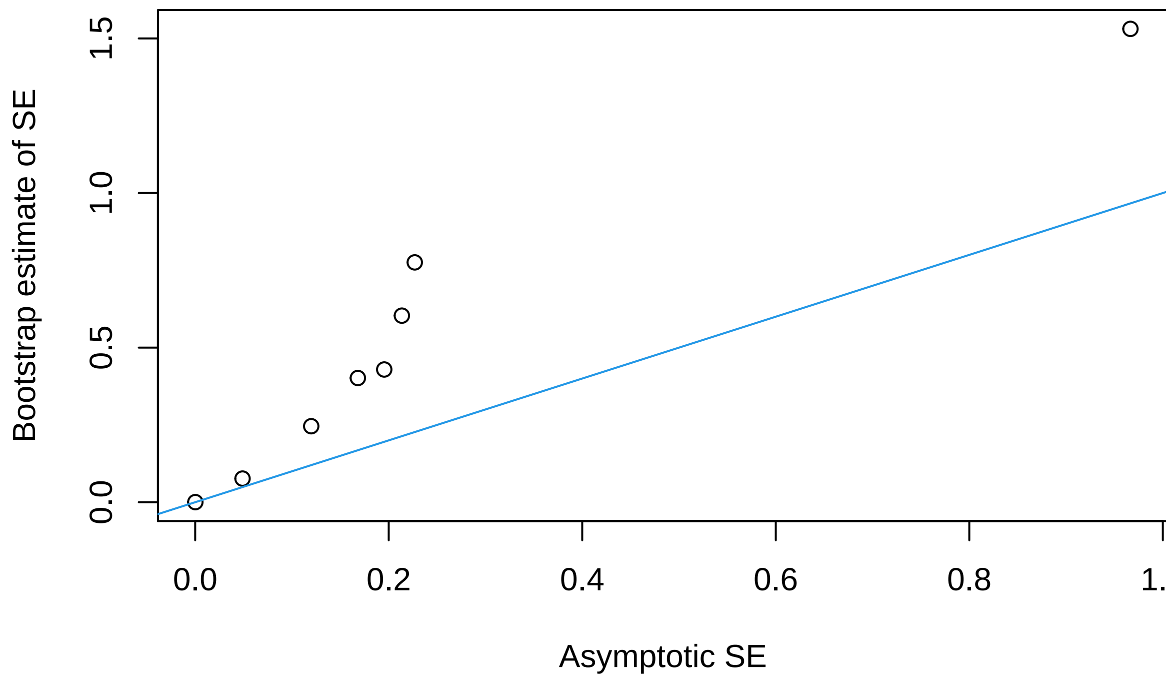
```
## For comparsion, here are the asymptotic SEs
```

```
 summary(fmGLM)
```

```
##
## Call:
## glm(formula = nSatellites ~ color + spine + width + weight, family = poisson(link = log),
##     data = DATA)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0290  -1.8630  -0.5988   0.9331   4.9446
```

```
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.3618003  0.9665506  -0.374  0.70817
## color3      -0.2648512  0.1681107  -1.575  0.11515
## color4      -0.5137051  0.1953624  -2.629  0.00855 **
## color5      -0.5308601  0.2269157  -2.339  0.01931 *
## spine2      -0.1503718  0.2135754  -0.704  0.48139
## spine3       0.0872826  0.1199287   0.728  0.46674
## width        0.0167487  0.0489197   0.342  0.73207
## weight       0.0004965  0.0001663   2.986  0.00283 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 632.79  on 172  degrees of freedom
## Residual deviance: 549.59  on 165  degrees of freedom
## AIC: 920.88
##
## Number of Fisher Scoring iterations: 6
```

```r
plot(SE~summary(fmGLM)$coef[,2],xlab='Asymptotic SE',ylab='Bootstrap estimate of SE')
abline(a=0,b=1,col=4)
```



Conclusions

Bootstrap suggest larger SE than the asymptotic SEs computed in 2.2. The Bootstrap CIs all include zero, except for weight. Thus, we only have evidence supporting the existence of an effect for weight.