

# - Assumptions -

*Ethan Shen and Steven Herrera*

*Due: December 2, 2018*

Given our model, below are the assumptions and model assessment features that we will cover.

Assumptions:

- 1) Plot of binned residuals vs. predicted values
- 2) Plot of binned residuals vs. numeric explanatory variables
- 3) Influential points and multicollinearity

Model Fit:

- 1) Examine confusion matrix
- 2) Examine ROC curve

## Final Model

Below, is our final model with interaction effects.

```
final.base.model <- model.selected.interactions  
kable(tidy(final.base.model), format = "markdown", digits = 3)
```

term	estimate	std.error	statistic	p.value
(Intercept)	10.392	3.518	2.954	0.003
minutes	-0.008	0.003	-2.844	0.004
ht	-0.048	0.019	-2.547	0.011
rankpoints	0.000	0.000	5.441	0.000
ace	0.110	0.040	2.764	0.006
df	-0.243	0.070	-3.474	0.001
bpSaved	-0.075	0.029	-2.548	0.011
surfaceGrass	5.285	8.299	0.637	0.524
surfaceHard	-8.048	4.294	-1.874	0.061
ht:surfaceGrass	-0.043	0.045	-0.940	0.347
ht:surfaceHard	0.040	0.023	1.710	0.087
ace:surfaceGrass	0.164	0.080	2.048	0.041
ace:surfaceHard	-0.021	0.045	-0.469	0.639
df:surfaceGrass	0.425	0.132	3.205	0.001
df:surfaceHard	0.107	0.082	1.297	0.195

## Assumptions

### Binned Plots with Residuals vs Predicted

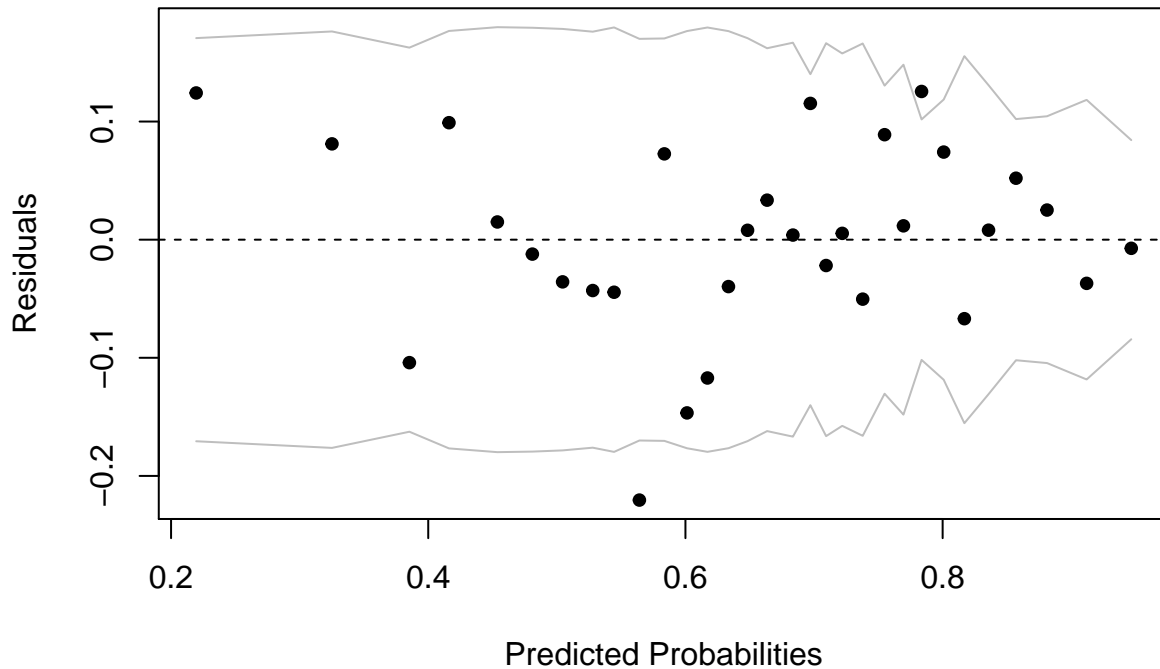
```
ten <- ten %>% mutate(Residuals = residuals.glm(final.base.model,type="response"),  
                      Predicted = predict.glm(final.base.model,type="response"))
```

```

binnedplot(ten$Predicted, ten$Residuals,xlab="Predicted Probabilities",
           ylab="Residuals",main="Binned Residuals vs. Predicted Probabilities")

```

## Binned Residuals vs. Predicted Probabilities



Looking at this plot, we do not see any violations of the assumptions. We see a plot that does not have a distinct pattern.

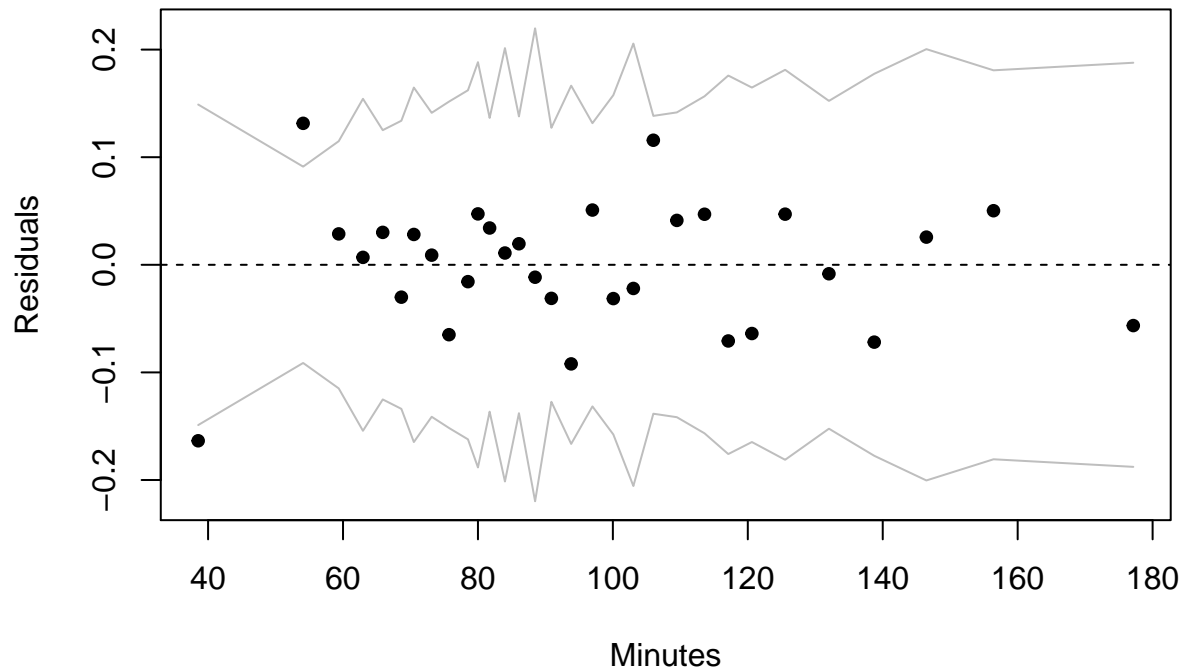
## Binned Residuals vs Numeric Explanatory Variables

```

binnedplot(ten$minutes, ten$Residuals,xlab="Minutes",
           ylab="Residuals",main="Binned Residuals vs. Minutes")

```

### Binned Residuals vs. Minutes

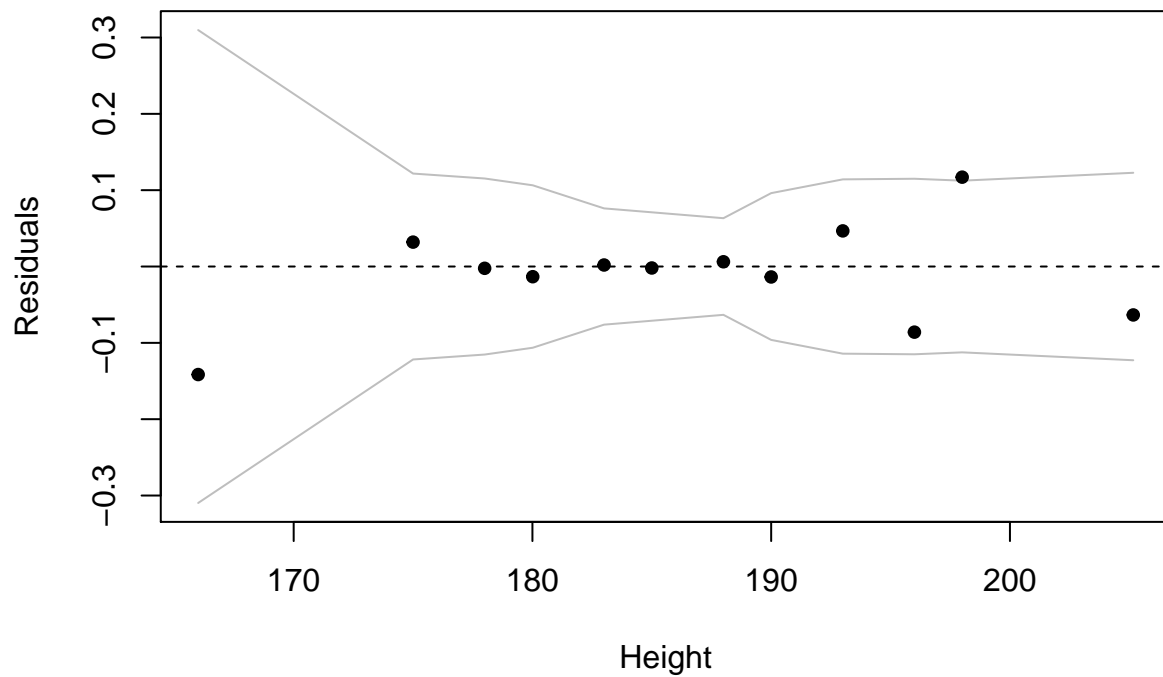


```

binnedplot(ten$ht, ten$Residuals,xlab="Height",
           ylab="Residuals",main="Binned Residuals vs. Height")

```

### Binned Residuals vs. Height

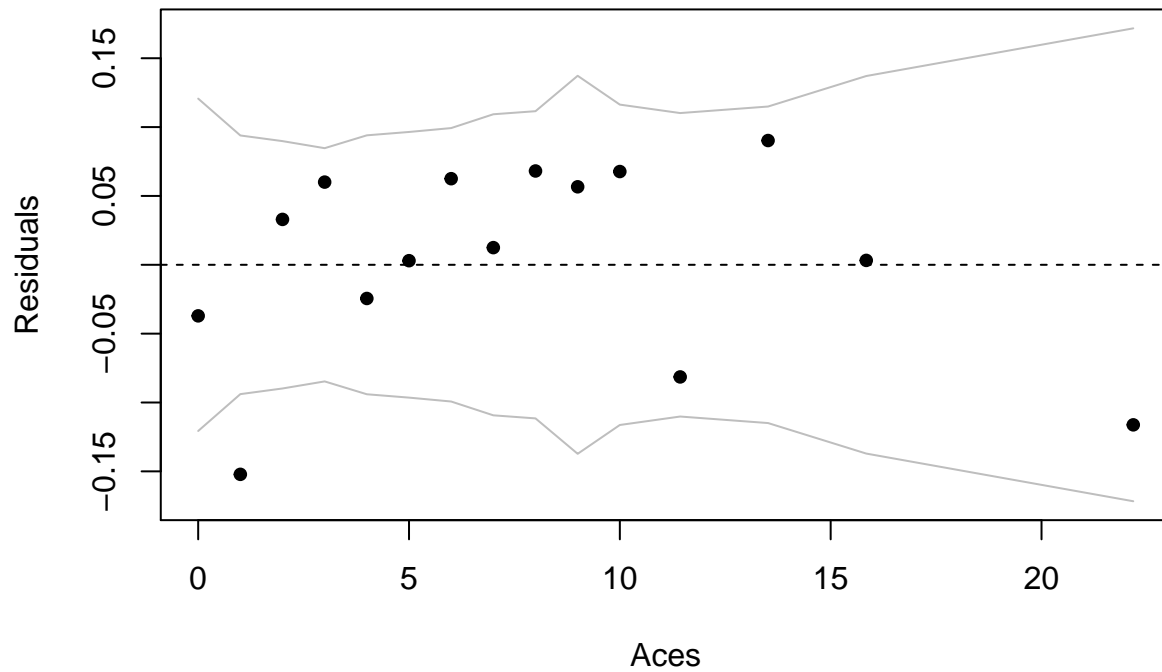


```

binnedplot(ten$ace, ten$Residuals,xlab="Aces",
           ylab="Residuals",main="Binned Residuals vs. Aces")

```

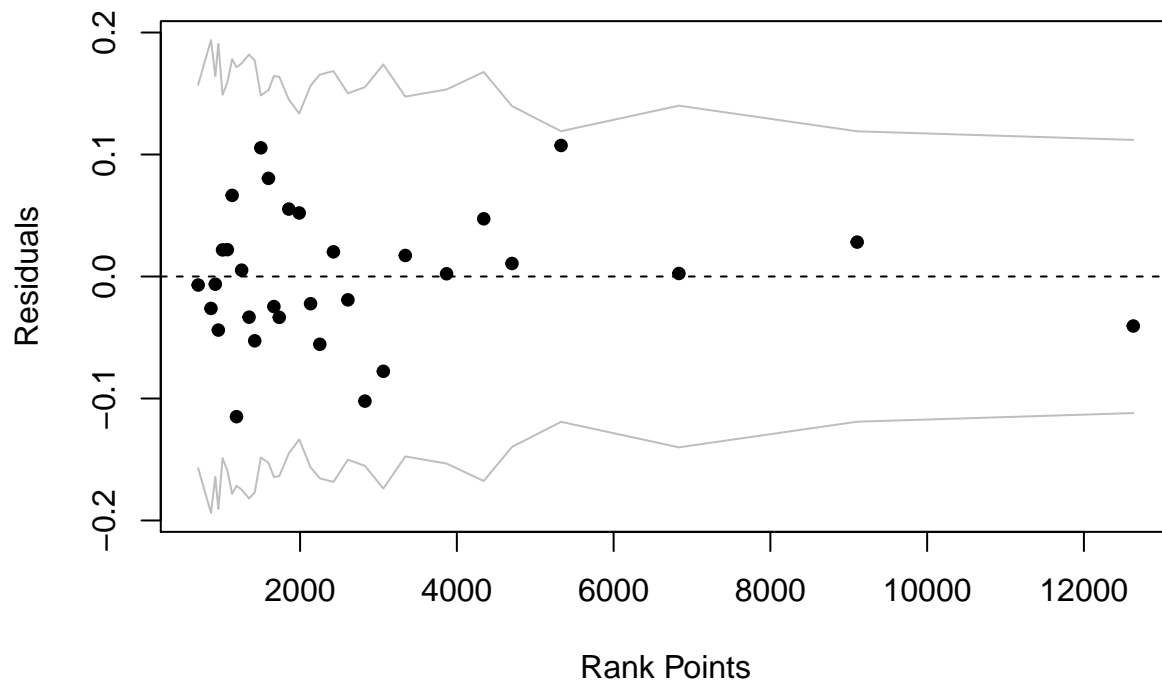
### Binned Residuals vs. Aces



```

binnedplot(ten$rankpoints, ten$Residuals,xlab="Rank Points",
           ylab="Residuals",main="Binned Residuals vs. Rank Points")
    
```

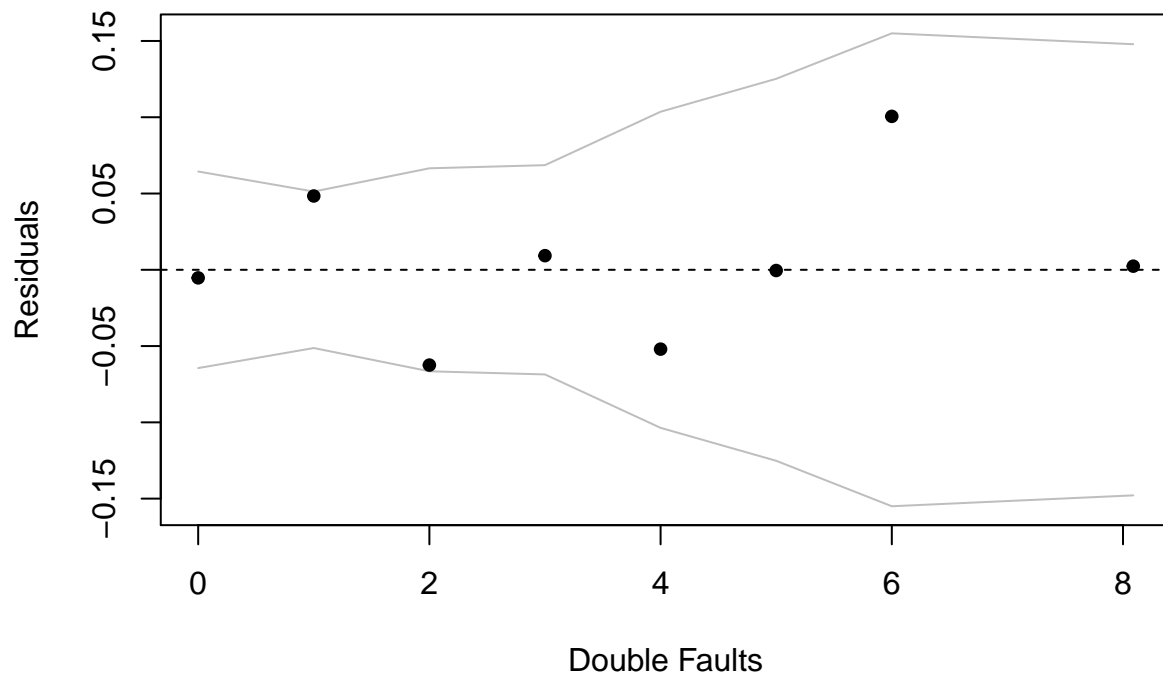
### Binned Residuals vs. Rank Points



```

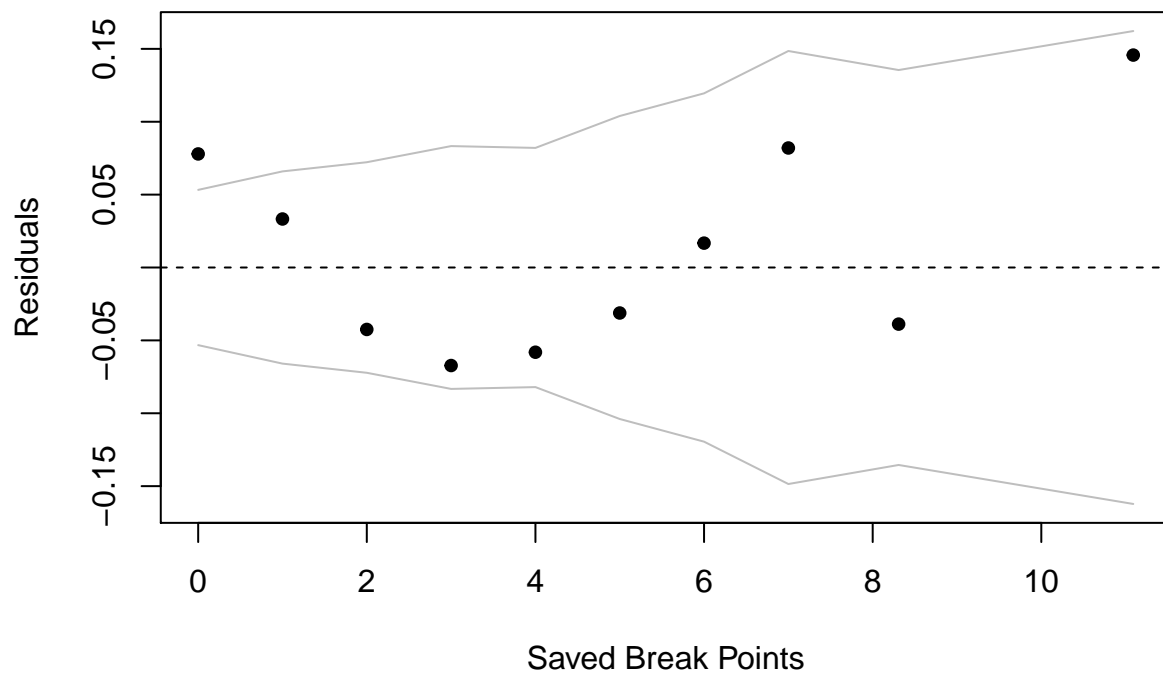
binnedplot(ten$df, ten$Residuals,xlab="Double Faults",
           ylab="Residuals",main="Binned Residuals vs. Double Faults")
    
```

### Binned Residuals vs. Double Faults



```
binplot(ten$bpSaved, ten$Residuals,xlab="Saved Break Points",
        ylab="Residuals",main="Binned Residuals vs. Saved Break Points")
```

### Binned Residuals vs. Saved Break Points



Looking at the binned residual plots, we see that all of the plots except for the binned residuals vs. saved break point have random scatter. The binned residuals vs. saved break point shows a pattern. This is a violation

of the assumptions.

## Influential Points

### VIF

```
tidy(vif(final.base.model))

## Warning: 'tidy.matrix' is deprecated.
## See help("Deprecated")

## # A tibble: 10 x 4
##   .rownames      GVIF      Df GVIF..1..2.Df..
##   <chr>         <dbl> <dbl>         <dbl>
## 1 minutes       1.45      1           1.21
## 2 ht            3.99      1           2.00
## 3 rankpoints    1.04      1           1.02
## 4 ace           6.82      1           2.61
## 5 df            4.02      1           2.00
## 6 bpSaved       1.32      1           1.15
## 7 surface     650920.      2           28.4
## 8 ht:surface   721588.      2           29.1
## 9 ace:surface   43.8       2           2.57
## 10 df:surface  14.7       2           1.96
```

After looking at the VIF values, we see that the VIF for **surface** is greater than 10, so we will also remove it from the model. This means we will also have to remove the interaction variables as well.

## Linear Regression Assumptions: Revised

Because one of our residuals plots has a non-linear relationship, we will remove **bpSaved** from the model and redo the assumptions. We will also remove **surface** and its corresponding interactions effects.

```
newten <- ten
final <- glm(status ~ minutes + ht + rankpoints + ace + df, family = binomial,
             data = newten)
kable(tidy(final), format = "markdown", digits = 6)
```

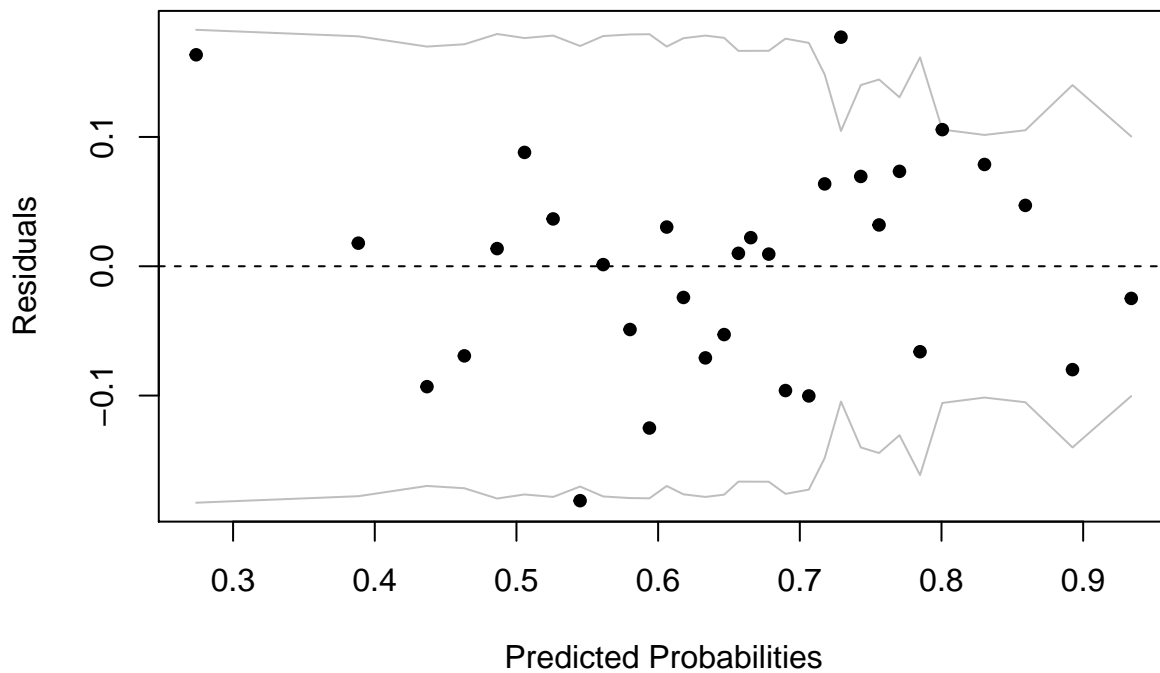
term	estimate	std.error	statistic	p.value
(Intercept)	5.197141	1.950891	2.663983	0.007722
minutes	-0.008897	0.002395	-3.714847	0.000203
ht	-0.023129	0.010416	-2.220635	0.026376
rankpoints	0.000184	0.000035	5.237843	0.000000
ace	0.094100	0.017519	5.371298	0.000000
df	-0.169342	0.035503	-4.769779	0.000002

## Model Assessment

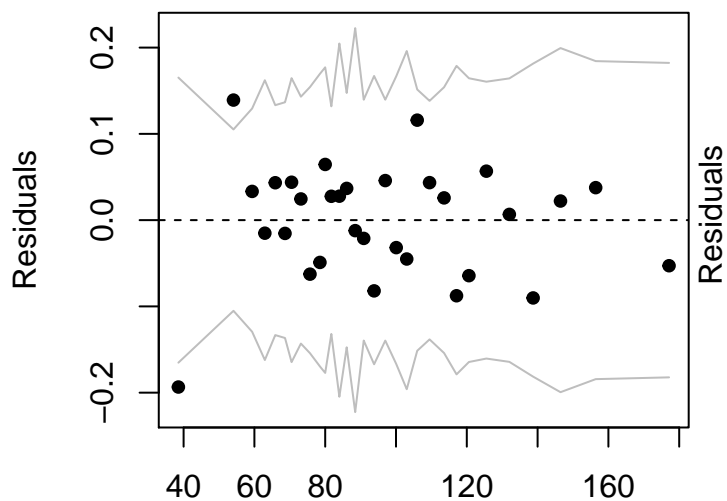
### Binned Plots with Residuals vs Predicted

```
newten <- newten %>% mutate(Residuals = residuals.glm(final,type="response"),  
                           Predicted = predict.glm(final,type="response"))  
  
binnedplot(newten$Predicted, newten$Residuals,xlab="Predicted Probabilities",  
           ylab="Residuals",main="Binned Residuals vs. Predicted Probabilities")
```

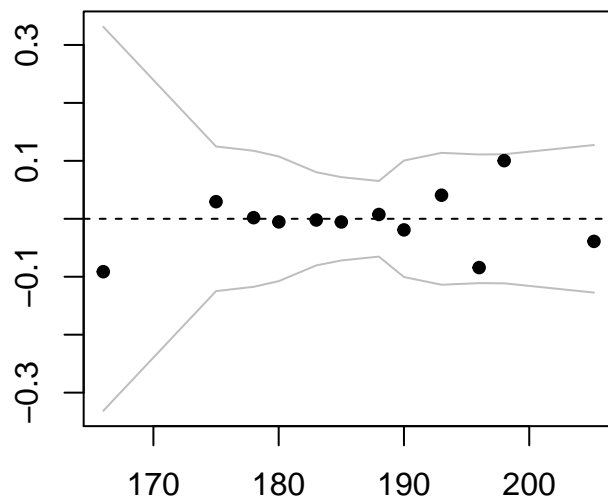
### Binned Residuals vs. Predicted Probabilities



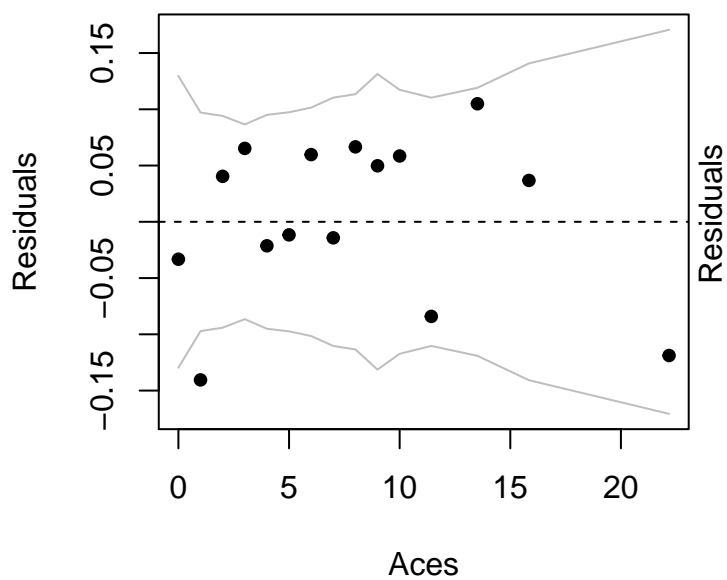
**Binned Residuals vs. Minutes**



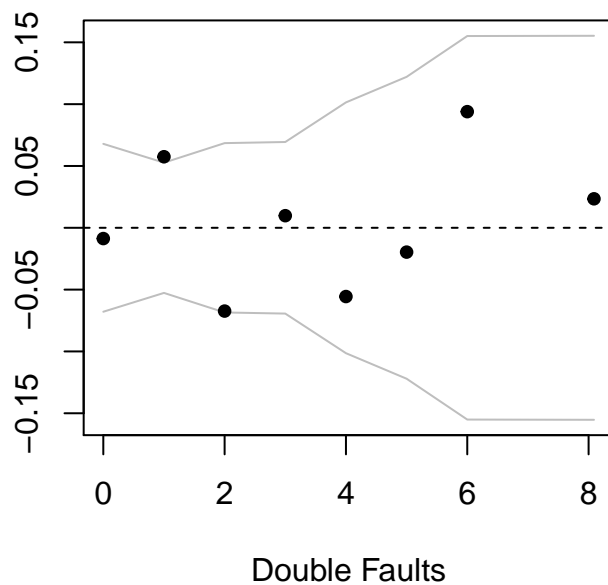
**Binned Residuals vs. Height**



**Binned Residuals vs. Aces**

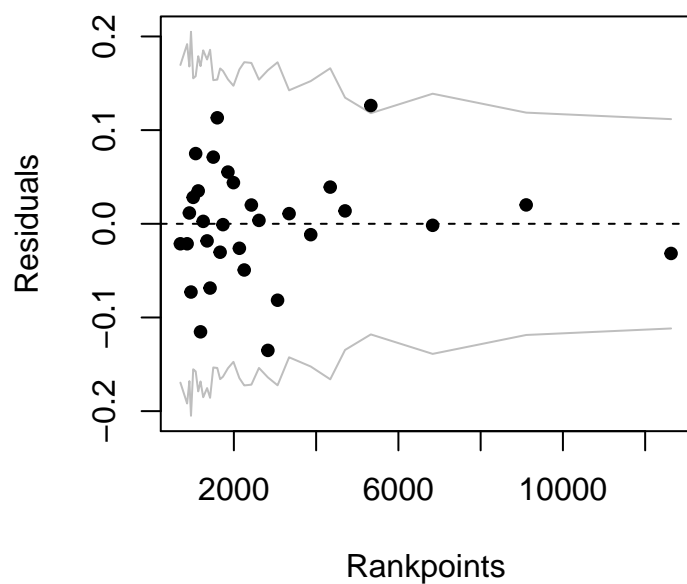


**Binned Residuals vs. Double Faults**

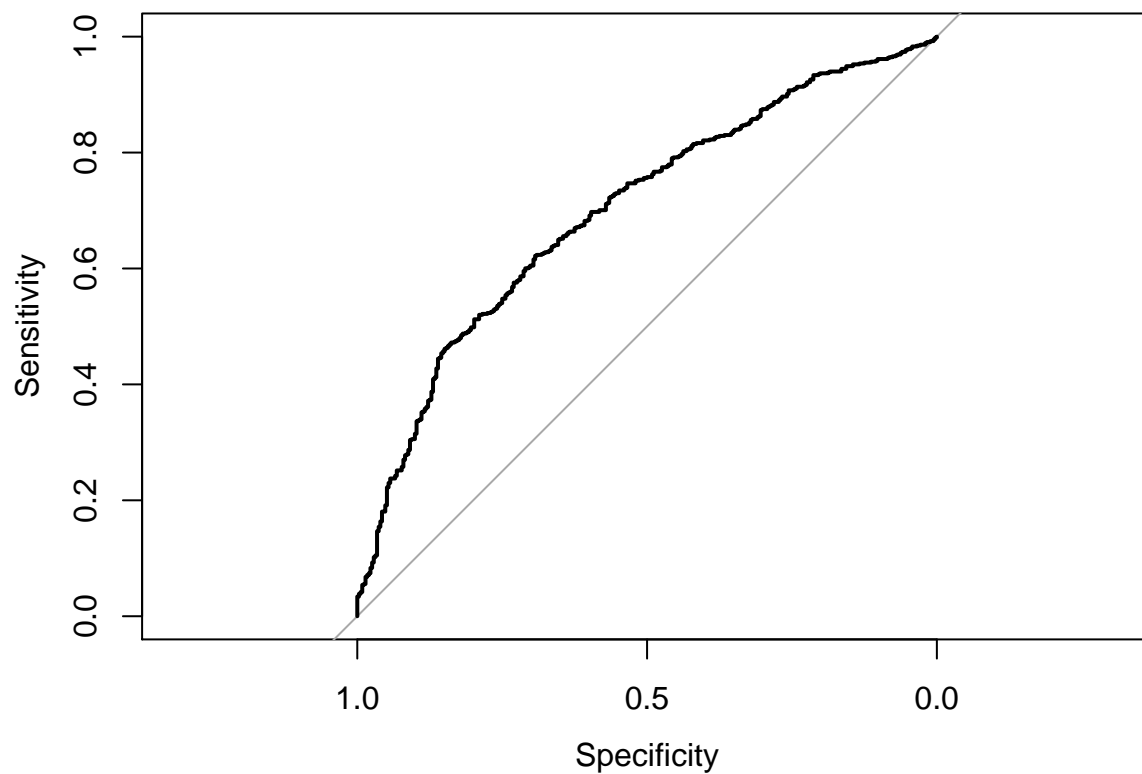




## Binned Residuals vs. Rankpoints



```
ROC.newten <- roc(newten$status,newten$Predicted,plot=T)
```



```
ROC.newten$auc
```

```
## Area under the curve: 0.6996
```

```
threshold = 0.30
```

```
table(newten$status, newten$Predicted > threshold)
```

```
##
##      FALSE TRUE
##    0      10  342
##    1      10  638

(342 + 10)/(342 + 10 + 10 + 638)

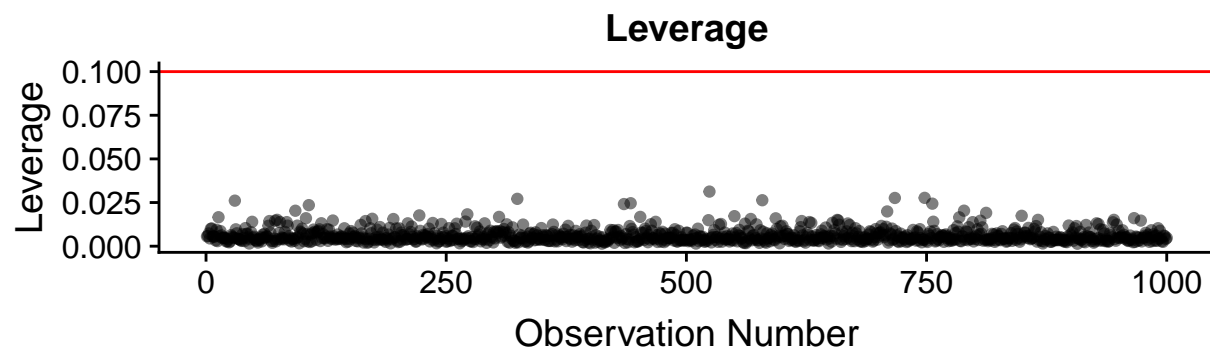
## [1] 0.352
```

## Influential Points

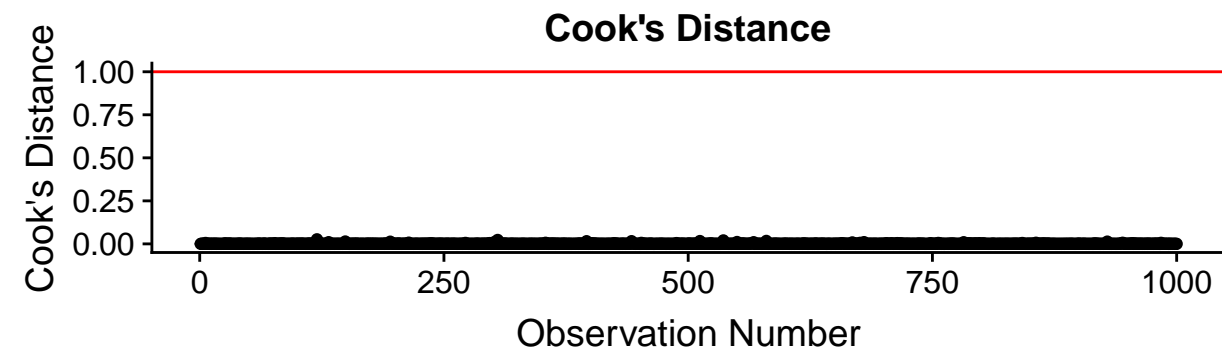
```
newten <- newten %>%
  mutate(leverage = hatvalues(final),
         cooks = cooks.distance(final),
         stand.resid = rstandard(final),
         obs.num = row_number())
```

### Leverage and Cook's Distance

```
ggplot(data=newten, aes(x=obs.num,y=leverage)) +
  geom_point(alpha=0.5) +
  geom_hline(yintercept=0.1,color="red")+
  labs(x="Observation Number",y="Leverage",title="Leverage")
```



```
ggplot(data=newten, aes(x=obs.num,y=cooks)) +
  geom_point() +
  geom_hline(yintercept=1,color="red")+
  labs(x="Observation Number",y="Cook's Distance",title="Cook's Distance")
```



## VIF

```
tidy(vif(final))
```

```
## Warning: 'tidy.numeric' is deprecated.  
## See help("Deprecated")  
  
## # A tibble: 5 x 2  
##   names      x  
##   <chr>    <dbl>  
## 1 minutes  1.19  
## 2 ht       1.28  
## 3 rankpoints 1.02  
## 4 ace      1.41  
## 5 df       1.08
```

## Conclusion

With VIF values less than 10, and binned residual plots that complete the assumptions, we have cleared model assessment and assumptions for the following final model:

```
kable(tidy(final), format = "markdown", digits = 6)
```

term	estimate	std.error	statistic	p.value
(Intercept)	5.197141	1.950891	2.663983	0.007722
minutes	-0.008897	0.002395	-3.714847	0.000203
ht	-0.023129	0.010416	-2.220635	0.026376
rankpoints	0.000184	0.000035	5.237843	0.000000
ace	0.094100	0.017519	5.371298	0.000000
df	-0.169342	0.035503	-4.769779	0.000002