STOR 590 HW4 Solution

Taebin Kim

Page 172 Exercise 4

Part (a)

We fit a Poisson model to the species response with the five geographic variables as predictors. We report the values of the coefficients and the deviance.

```
data(gala, package = "faraway")
glm1 <- glm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
family = poisson, gala)
coef(glm1)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223
##      Adjacent
## -0.0006630311
```

```
deviance(glm1)
```

```
## [1] 716.8458
```

Part (b)

For a Poisson GLM, we have $\eta = \log(\mu)$, $\frac{d\eta}{d\mu} = \frac{1}{\mu}$, $V(\mu) = \mu$, and $w = \mu$. The adjusted dependent variable has the form $z = \eta + \frac{y-\mu}{\mu}$.

Part (c)

We use the observed response as initial values to compute the first stage of the iteration, stopping right after the first linear model fit.

```
y <- gala$Species
mu <- y
eta <- log(mu)
z <- eta + (y - mu) / mu
w <- mu
lm1 <- lm(z ~ Area + Elevation + Nearest + Scruz + Adjacent, weights = w, gala)
coef(lm1)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.5191545412 -0.0005298484  0.0031643557  0.0025188990 -0.0037899780
##      Adjacent
## -0.0006623523
```

The coefficients of the first linear model fit are already pretty close to those produced by the GLM fit.

Part (d)

We continue the iteration to get the next η and μ. We use this to compute the current value of the deviance.

```
eta <- fitted(lm1)
mu <- exp(eta)
deviance <- 2 * sum(y * log(y / mu) - (y - mu))
deviance
```

```
## [1] 828.0096
```

Part (e)

We compute one more iteration of the GLM fit and report the next calculation of the coefficients and the deviance.

```
z <- eta + (y - mu) / mu
w <- mu
lm1 <- lm(z ~ Area + Elevation + Nearest + Scruz + Adjacent, weights = w, gala)
coef(lm1)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.2102594447 -0.0005651969  0.0034606226  0.0077171134 -0.0052400871
##      Adjacent
## -0.0006604828
```

```
eta <- fitted(lm1)
mu <- exp(eta)
deviance1 <- 2 * sum(y * log(y / mu) - (y - mu))
deviance1
```

```
## [1] 719.4158
```

The coefficients and devaince of the second linear model fit are even closer to those produced by the GLM fit.

Part (f)

We repeat these iterations a few more times, computing the deviance in each time. We stop when the deviance does not change much.

```
while (abs(deviance1 - deviance) > 1e-9) {
deviance <- deviance1
z <- eta + (y - mu) / mu
w <- mu
lm1 <- lm(z ~ Area + Elevation + Nearest + Scruz + Adjacent, weights = w, gala)
eta <- fitted(lm1)
mu <- exp(eta)
deviance1 <- 2 * sum(y * log(y / mu) - (y - mu))
}
coef(lm1)
```

```
##   (Intercept)          Area     Elevation       Nearest         Scruz
##  3.1548078779 -0.0005799429  0.0035405940  0.0088255719 -0.0057094223
##      Adjacent
## -0.0006630311
```

The final estimated coefficients are exactly the same as those produced by the GLM fit.

Part (g)

We use the final iterated linear model fit to produce standard errors for the coefficients.

```
summary(lm1)$coef[, 2] / summary(lm1)$sigma
```

```
##  (Intercept)         Area    Elevation      Nearest        Scruz
## 5.174955e-02 2.627299e-05 8.740709e-05 1.821261e-03 6.256214e-04
##     Adjacent
## 2.932754e-05
```

```
summary(glm1)$coef[, 2]
```

```
##  (Intercept)         Area    Elevation      Nearest        Scruz
## 5.174952e-02 2.627298e-05 8.740704e-05 1.821260e-03 6.256200e-04
##     Adjacent
## 2.932754e-05
```

Page 173 Exercise 6

Part (a)

We fit a Poisson model with the number of shots as the response and team, position, tackles and passes per game as predictor.

```
data(worldcup, package = "faraway")
# remove goalkeepers
worldcup1 <- worldcup[worldcup$Position != "Goalkeeper", ]
# compute new variables representing tackles/passes per 90-minute game
worldcup1$Tackles90 <- worldcup1$Tackles / worldcup1$Time * 90
worldcup1$Passes90 <- worldcup1$Passes / worldcup1$Time * 90
glm1 <- glm(Shots ~ offset(log(Time)) + Team + Position + Tackles90 + Passes90,
family = poisson, worldcup1)
coef(glm1)[c("Tackles90", "Passes90")]
```

```
##     Tackles90      Passes90
## -0.0873869727  0.0007915914
```

We can see that tackles have a negative association with shots, whereas passes have a positive association with shots.

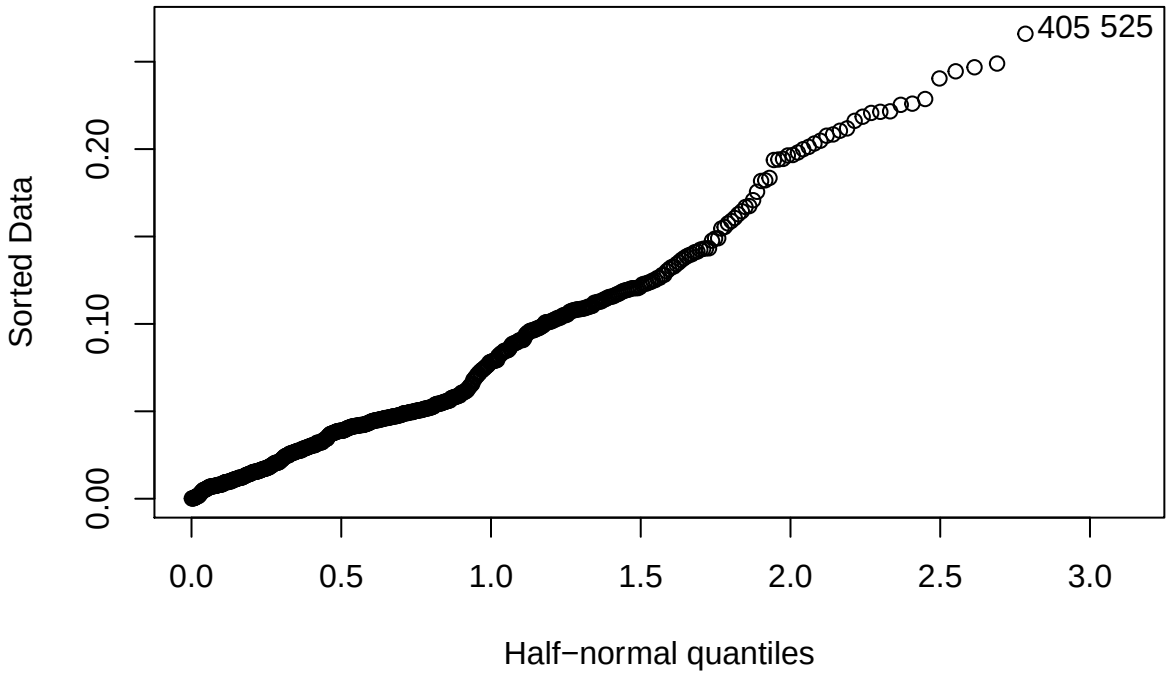
Part (b)

We calculate the leverages for the current model and make an appropriate plot of the leverages.

```
library(faraway)
leverages <- influence(glm1)$hat
worldcup1[which.max(leverages), ]
```

```
##        Team Position Time Shots Passes Tackles Saves Tackles90 Passes90
## Villa Spain  Forward  529    22    169       2     0 0.3402647 28.75236
```

```
halfnorm(leverages)
```



It shows that Villa has the highest leverage. This might be because Spain won the 2010 World Cup and Villa was one of Spain's forward player who made a lot of shots. The plot shows no evidence of any leverage that is exceptional.

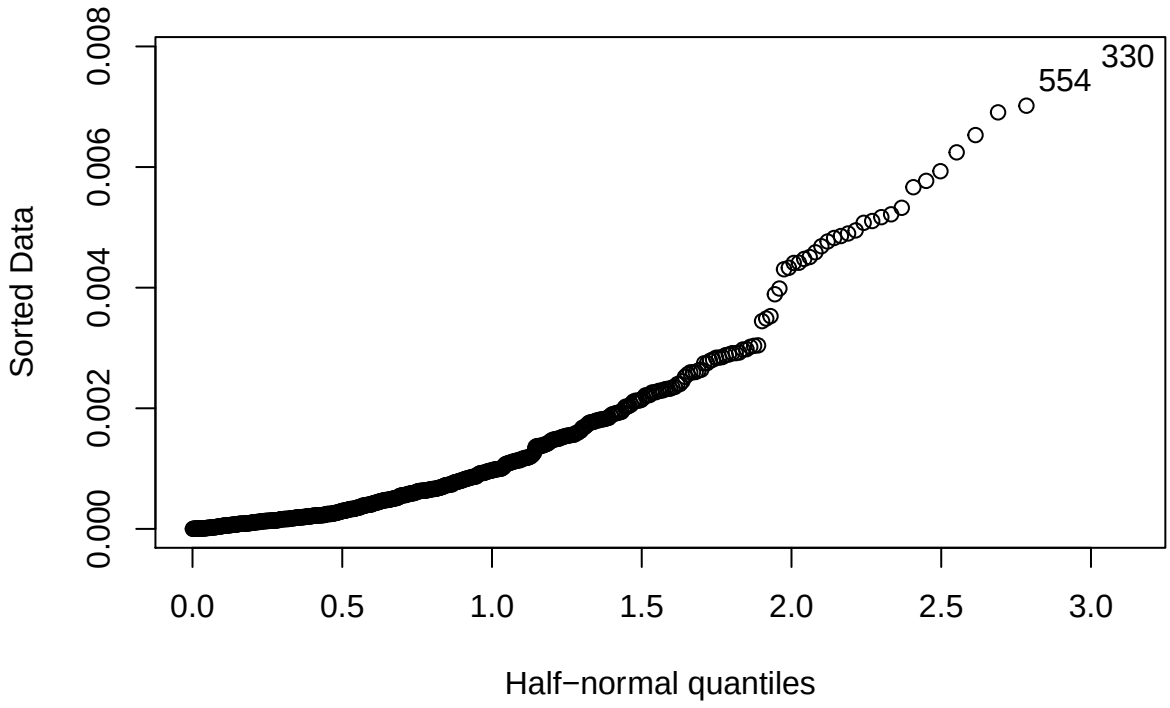
Part (c)

We compute the change in the regression coefficients when each case is dropped and make an appropriate plot of the change in the tackle coefficient.

```
DFBETAS <- influence(glm1)$coef
worldcup1[which.max(abs(DFBETAS[, "Tackles90"])), ]
```

```
##                 Team   Position Time Shots Passes Tackles Saves Tackles90
## Mascherano Argentina Midfielder  360     0    237      19     0      4.75
##            Passes90
## Mascherano    59.25
```

```
halfnorm(abs(DFBETAS[, "Tackles90"]))
```



We can observe that dropping Mascherano causes the greatest absolute change in the tackle coefficient. Interestingly, Mascherano did not make any shots during the 2010 World Cup. Again, the plot shows no evidence of any value that is particularly large.

Part (d)

We calculate the Cook Statistics.

```
Cook <- cooks.distance(glm1)
worldcup1[which.max(Cook), ]
```

```
##         Team   Position Time Shots Passes Tackles Saves Tackles90 Passes90
## Dempsey  USA Midfielder  390    15    137       6     0  1.384615 31.61538
```

The result shows that Dempsey has the largest Cook statistic.

Part (e)

We find the jackknife residuals.

```
jackknife <- rstudent(glm1)
worldcup1[which.max(abs(jackknife)), ]
```

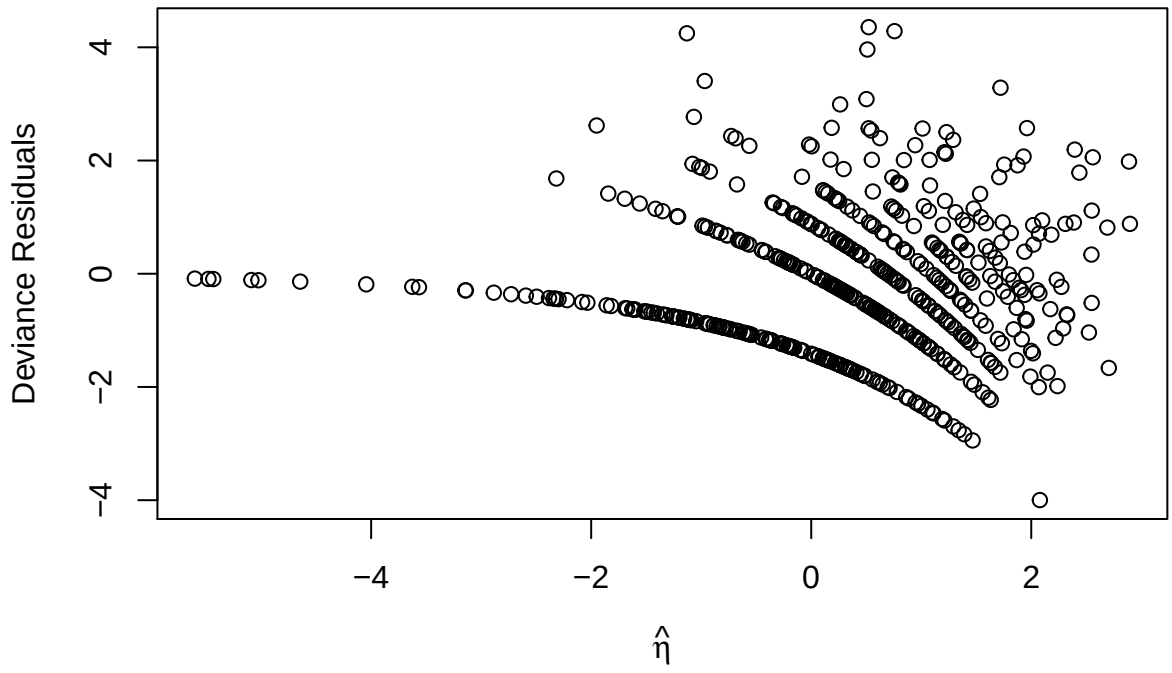
```
##            Team   Position Time Shots Passes Tackles Saves Tackles90
## GonzalezC Chile Midfielder  138    10     41       6     0  3.913043
##           Passes90
## GonzalezC 26.73913
```

We can see that GonzalezC has the largest absolute jackknife residual.

Part (f)

We plot the residuals against the appropriate fitted values.

```
plot(residuals(glm1) ~ predict(glm1, type = "link"),
xlab = expression(hat(eta)), ylab = "Deviance Residuals")
```

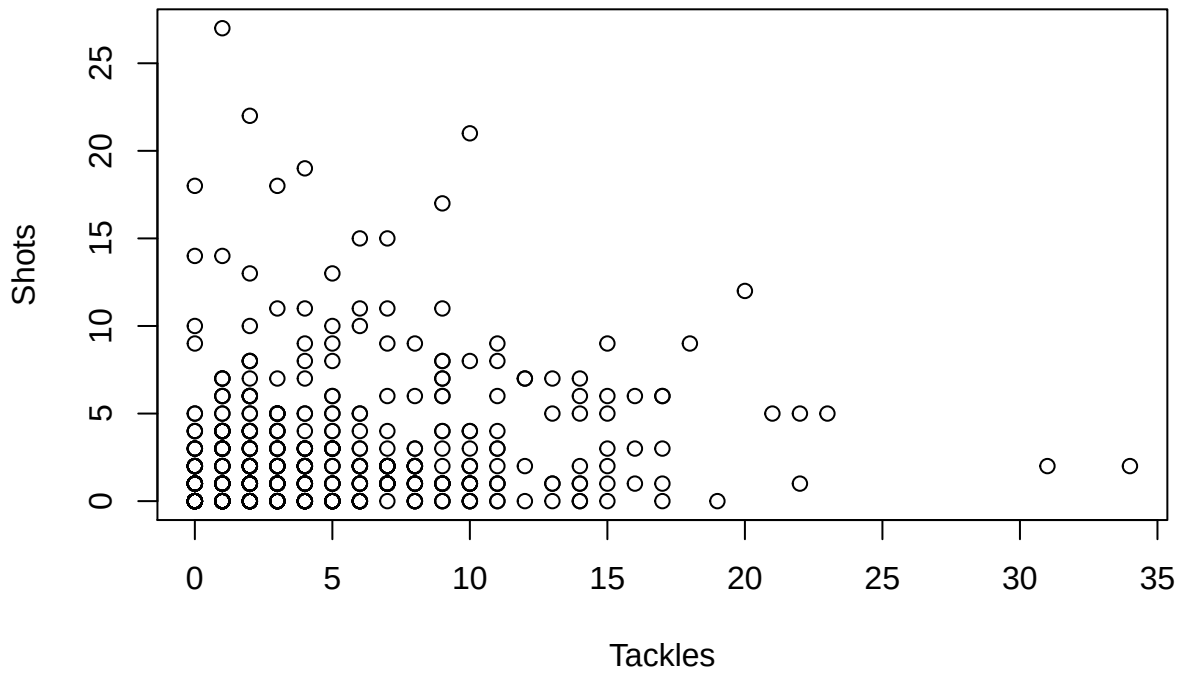


The plot of the residuals against the fitted values shows lines of points, because the response variable `Shots` represents counts and are mostly between 1 and 11. The plot indicates some signs of non-constant variance.

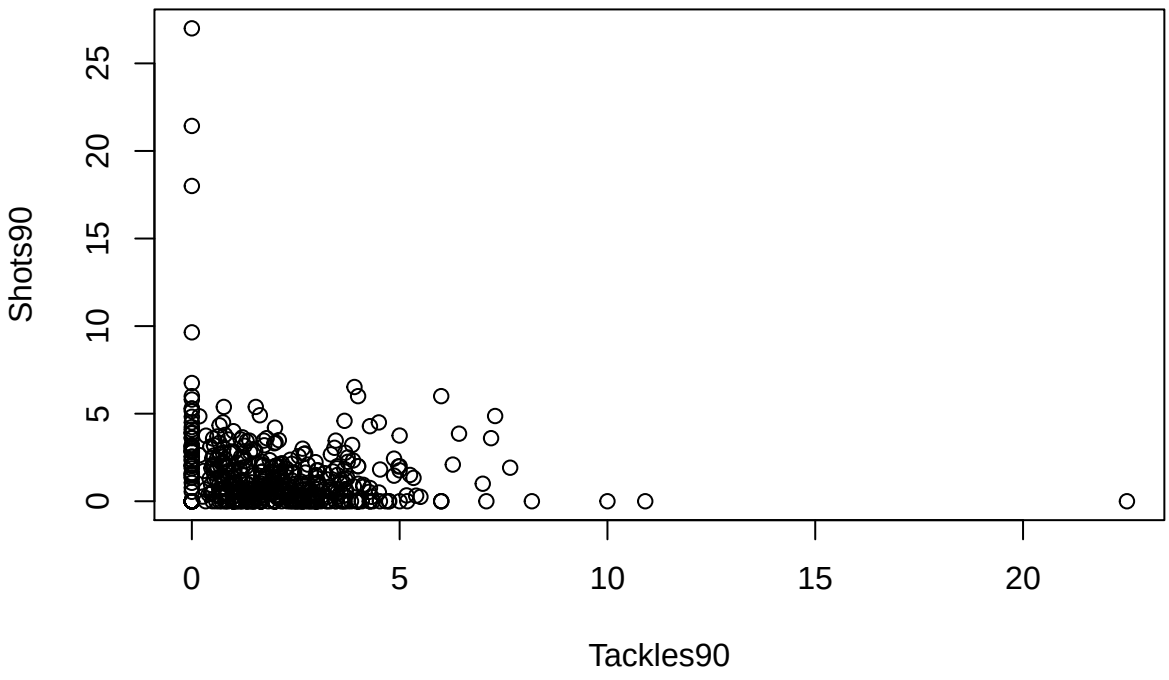
Part (g)

We make the following three plots:

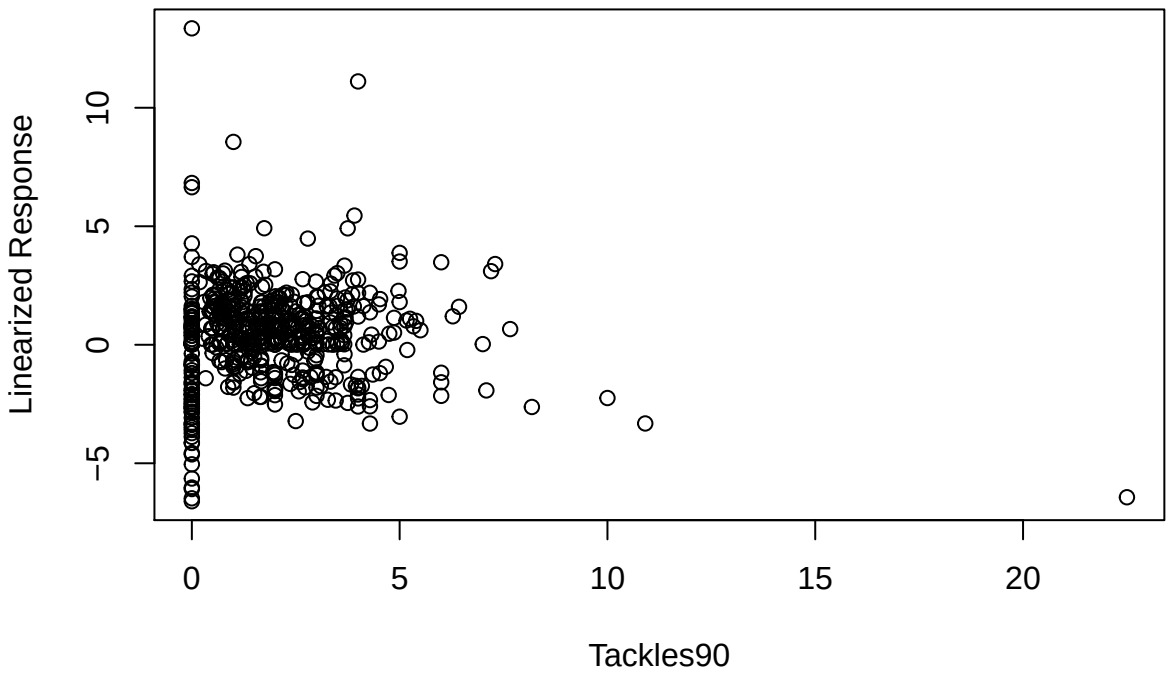
```
plot(Shots ~ Tackles, worldcup1)
```




```
worldcup1$Shots90 <- worldcup1$Shots / worldcup1$Time * 90
plot(Shots90 ~ Tackles90, worldcup1)
```




```
mu <- predict(glm1, type = "response")
z <- predict(glm1) + (worldcup1$Shots - mu) / mu
plot(z ~ Tackles90, worldcup1, ylab = "Linearized Response")
```

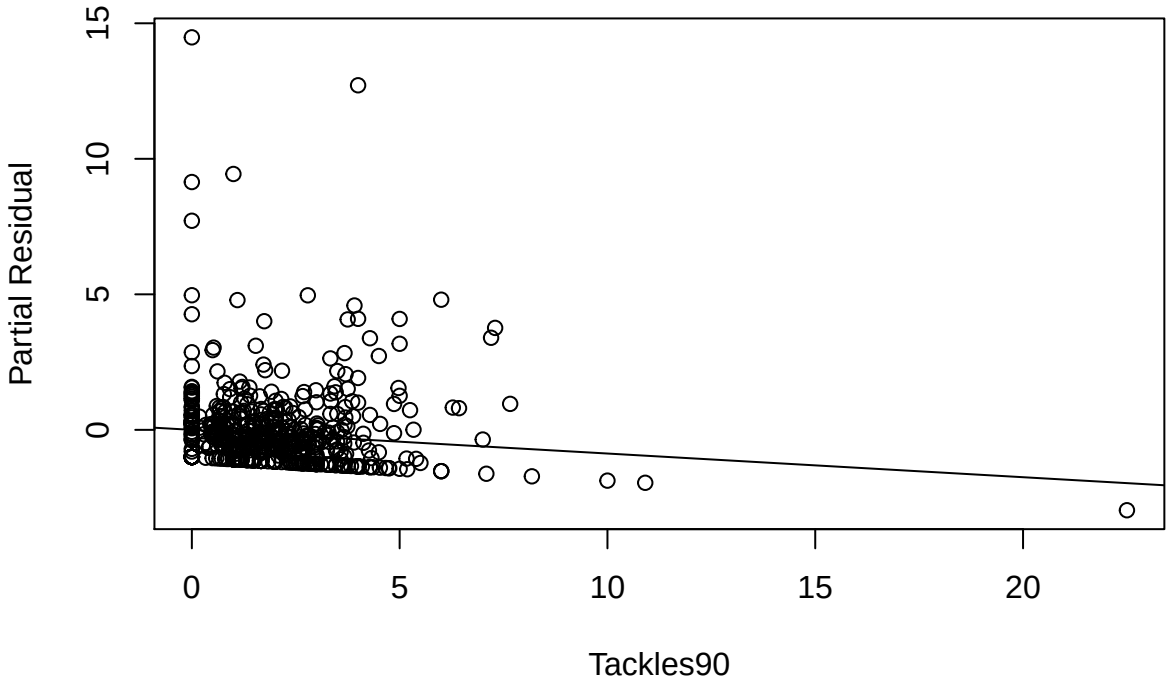


We choose the third plot for discovering the relationship between this predictor and the response, because only the third plot would show a linear relationship in the ideal case. In this case we observe a negative linear relationship, which agrees with the negative coefficient of tackles per game in our GLM model.

Part (h)

We construct the partial residual plot for tackles.

```
mu <- predict(glm1, type = "response")
u <- (worldcup1$Shots - mu) / mu + coef(glm1)["Tackles90"] * worldcup1$Tackles90
plot(u ~ Tackles90, worldcup1, ylab = "Partial Residual")
abline(0, coef(glm1)["Tackles90"])
```

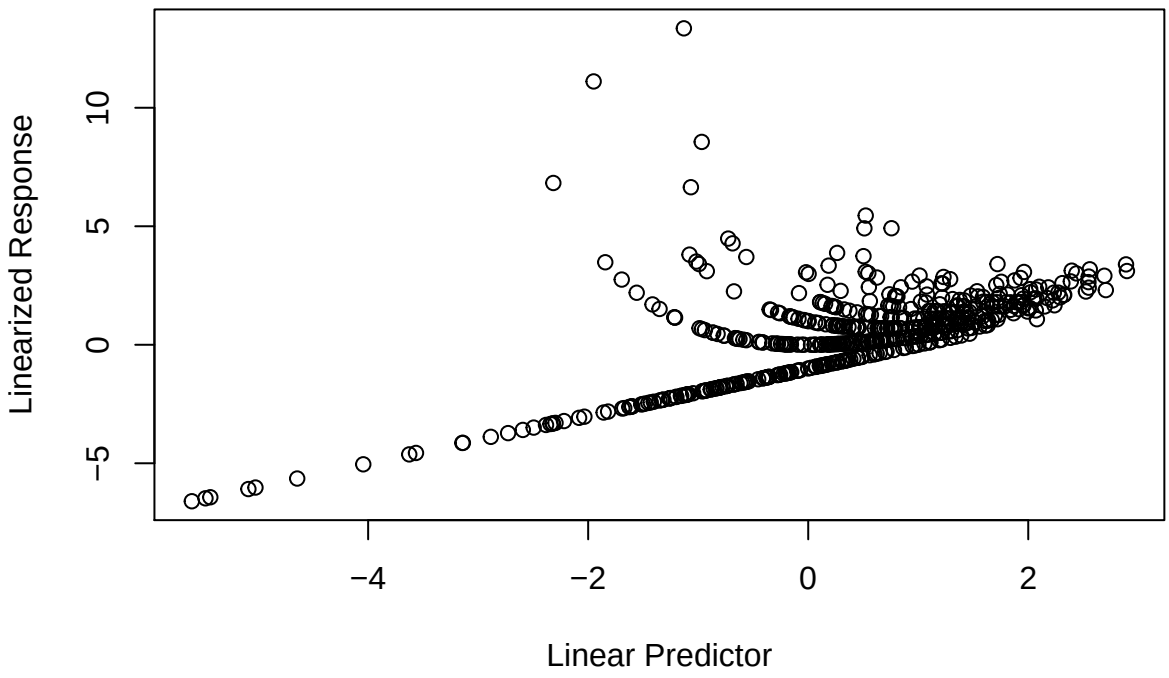


The partial residual plot also shows a negative linear relationship between shots and tackles per game. However, there are some points on the top left part of the plot that might require further checking. The point on the far right is not really influential, because it is close to the straight line.

Part (i)

We make a diagnostic plot to check the choice of the (default) link function.

```
z <- predict(glm1) + (worldcup1$Shots - mu) / mu
plot(z ~ predict(glm1), xlab = "Linear Predictor", ylab = "Linearized Response")
```



The diagnostic plot shows an overall linear relationship, although there are some points on the top center part of the plot that might require further checking.