Stats_CW

```
knitr::opts_chunk$set(
 results = "hold", echo = TRUE, eval = TRUE,
 message = FALSE, fig.width = 7,
 fig.height = 4, fig.align = "center"
)
```

```
library("tidyverse")
library("magrittr")
library("here")
library("janitor")
library("gridExtra")
library("readxl")
library("Lahman")
library("viridis")
library("lindia")
library("lme4")
library("caret")
library("pROC")
library("car")
```

Question 1a

```
# Create the df_managers dataset
df_managers <- Managers %>%
  clean_names() %>%
  mutate(win_pct = w / g) %>% # Calculate the proportion of games won
  select(player_id, team_id, year_id, lg_id, plyr_mgr, win_pct) # Select the required variables

glimpse(df_managers)
```

```
## Rows: 3,749
## Columns: 6
## $ player_id <chr> "wrighha01", "woodji01", "paborch01", "lennobi01", "deaneha0…
## $ team_id   <fct> BS1, CH1, CL1, FW1, FW1, NY2, PH1, RC1, TRO, TRO, WS3, BL1, …
## $ year_id   <int> 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871, 1871, …
## $ lg_id     <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, …
## $ plyr_mgr  <fct> Y, Y, Y, Y, Y, Y, Y, Y, Y, Y, N, Y, Y, Y, Y, Y, Y, Y, Y, Y, …
## $ win_pct   <dbl> 0.6451613, 0.6785714, 0.3448276, 0.3571429, 0.4000000, 0.484…
```

Question 1b

```
#Create df_teams with cleaned column names
df_teams <- Teams %>%
  clean_names() %>% #Standardise column names
  select(year_id, team_id, div_win, cs) #Select required variables

# Directly merge df_teams with df_managers to create man_teams
man_teams <- df_managers %>%
  left_join(df_teams, by = c("year_id", "team_id")) %>% # Merge datasets by year_id and team_id
  select(-lg_id) # Remove lg_id column

#Merge man_teams with AwardsShareManagers to create awards_man
awards_man <- AwardsShareManagers %>%
  clean_names() %>% # Ensure clean column names
  left_join(man_teams, by = c("player_id", "year_id")) # Merge datasets by player_id and year_id

#Add the new variable sqr_point_pct
awards_man <- awards_man %>%
  mutate(sqr_point_pct = sqrt(points_won / points_max)) # Calculate square root proportion

glimpse(awards_man)

#Clean the awards_man dataset
awards_man <- awards_man %>%
  drop_na() %>% # Remove rows with missing values
  mutate(team_id = droplevels(as.factor(team_id))) # Drop unused levels of team_id
```

```
## Rows: 512
## Columns: 13
## $ award_id      <chr> "BBWAA Manager of the Year", "BBWAA Manager of the Year"…
## $ year_id       <int> 1983, 1983, 1983, 1983, 1983, 1983, 1983, 1984, 1984, 19…
## $ lg_id         <fct> AL, AL, AL, NL, NL, NL, NL, AL, AL, AL, AL, AL, NL, NL, …
## $ player_id     <chr> "altobjo01", "coxbo01", "larusto01", "lasorto01", "lilli…
## $ points_won    <int> 7, 4, 17, 10, 9, 1, 4, 96, 9, 48, 95, 4, 101, 72, 2, 41,…
## $ points_max    <int> 28, 28, 28, 24, 24, 24, 24, 140, 140, 140, 140, 140, 120…
## $ votes_first   <int> 7, 4, 17, 10, 9, 1, 4, 13, 0, 4, 11, 0, 16, 4, 1, 4, 16,…
## $ team_id       <fct> BAL, TOR, CHA, LAN, HOU, PHI, PIT, DET, TOR, MIN, KCA, C…
## $ plyr_mgr      <fct> N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N,…
## $ win_pct       <dbl> 0.6049383, 0.5493827, 0.6111111, 0.5582822, 0.5246914, 0…
## $ div_win       <chr> "Y", "N", "Y", "Y", "N", "Y", "N", "Y", "N", "N", "Y", "…
## $ cs            <int> 33, 72, 50, 76, 95, 75, 77, 68, 67, 30, 64, 51, 66, 54, …
## $ sqr_point_pct <dbl> 0.50000000, 0.37796447, 0.77919372, 0.64549722, 0.612372…
```

Question 1c

```
# Fit the Gaussian model
spp_mod <- lm(sqr_point_pct ~ win_pct + div_win + cs, data = awards_man)

summary(spp_mod)
```

```
##
## Call:
## lm(formula = sqr_point_pct ~ win_pct + div_win + cs, data = awards_man)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52249 -0.18946 -0.03539  0.18009  0.66465
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.7278773  0.1409811  -5.163 3.56e-07 ***
## win_pct      1.8083774  0.2518894   7.179 2.67e-12 ***
## div_winY     0.1364128  0.0266417   5.120 4.42e-07 ***
## cs           0.0027219  0.0006249   4.356 1.62e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2401 on 482 degrees of freedom
## Multiple R-squared:  0.271,  Adjusted R-squared:  0.2665
## F-statistic: 59.74 on 3 and 482 DF,  p-value: < 2.2e-16
```

Fitted model

$$\text{sqr\_point\_pct} = -0.73 + 1.81 \cdot \text{win\_pct} + 0.14 \cdot \text{DivWin} + 0.0027 \cdot \text{CS}$$

Significance of predictors

All predictors (`win_pct` , `div_win` , and `cs`) have very small p-values (<0.001), meaning they are statistically significant at conventional levels. This indicates strong evidence that these variables influence sqr_point_pct.

Model fit

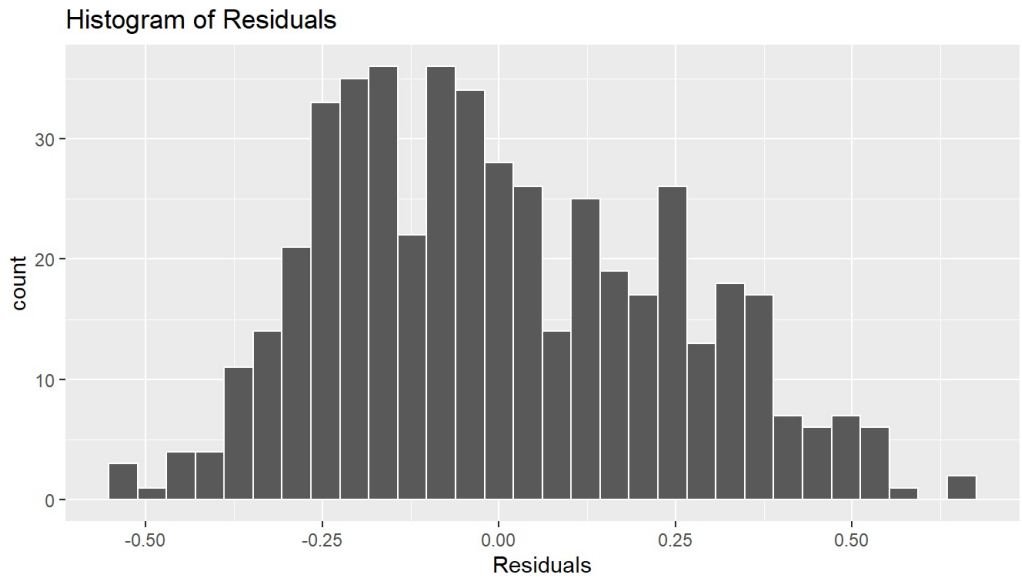
The range of sqr_point_pct is approximately 0.079 to 0.993, spanning 0.914 units. The Residual Standard Error (RSE) of 0.2401 indicates that the average deviation of the model's predictions from the actual values is about 26.3% of the total range, reflecting moderate accuracy. There is still a noticeable degree of error relative to the range of the outcome variable.

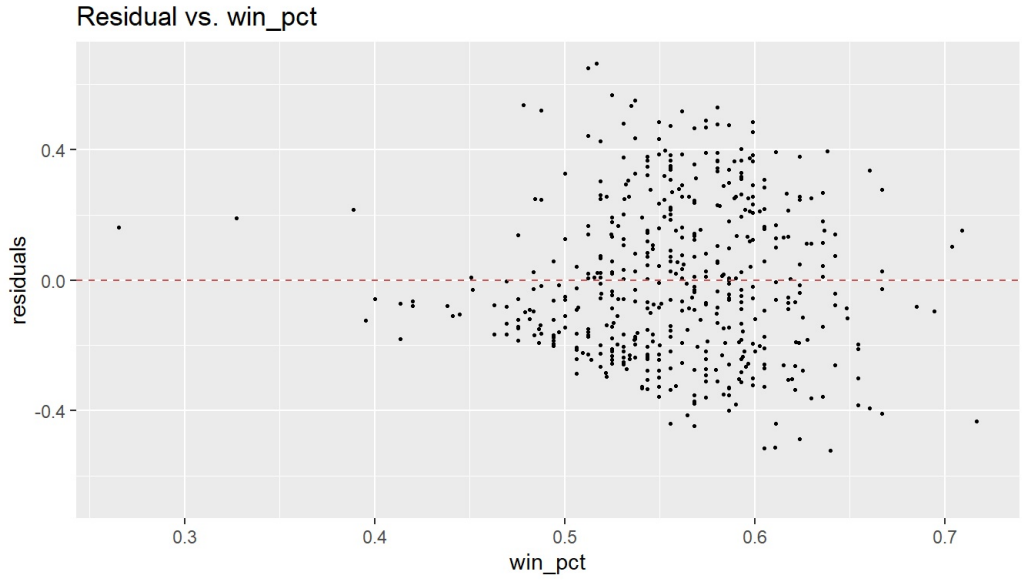
An R-squared of 0.271 means that only 27.1% of the variability in sqr_point_pct is accounted for by the model. This is a relatively low R-squared value, meaning the model explains only a modest amount of the variation in the outcome.There are likely other important factors influencing sqr_point_pct that are missing from the model.

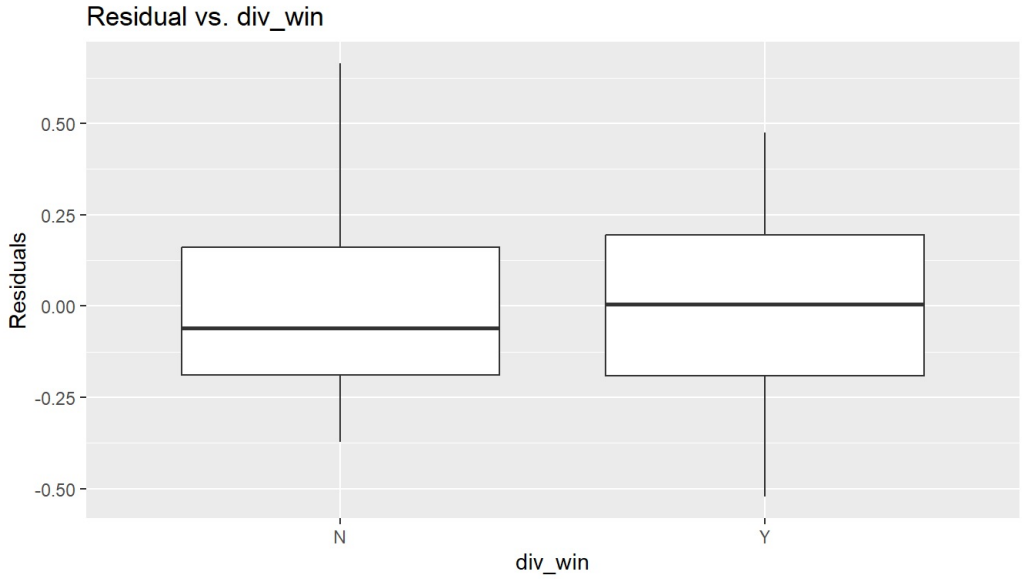
The very large F-statistic (59.74) and the extremely small p-value (< 2.2e-16) indicate that the model as a whole is statistically significant. In other words, at least one of the predictors contributes meaningfully to explaining the variability in sqr_point_pct.The significant F-statistic confirms that the predictors collectively improve the model over using just the intercept (mean of the response variable) alone. This is a good indication that the model has value, though it could benefit from adding more relevant predictors or interactions to capture additional variability.

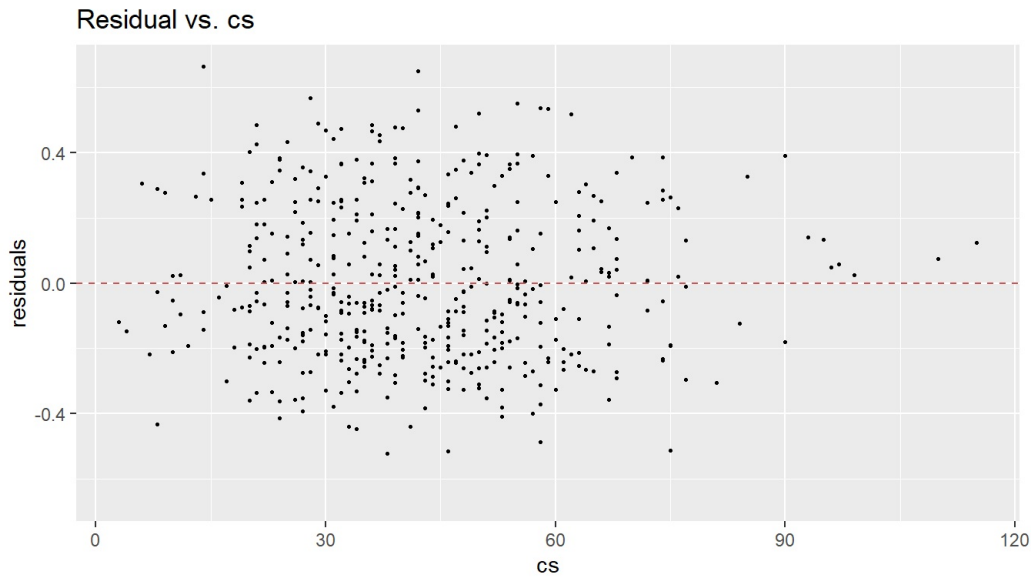
Question 1d

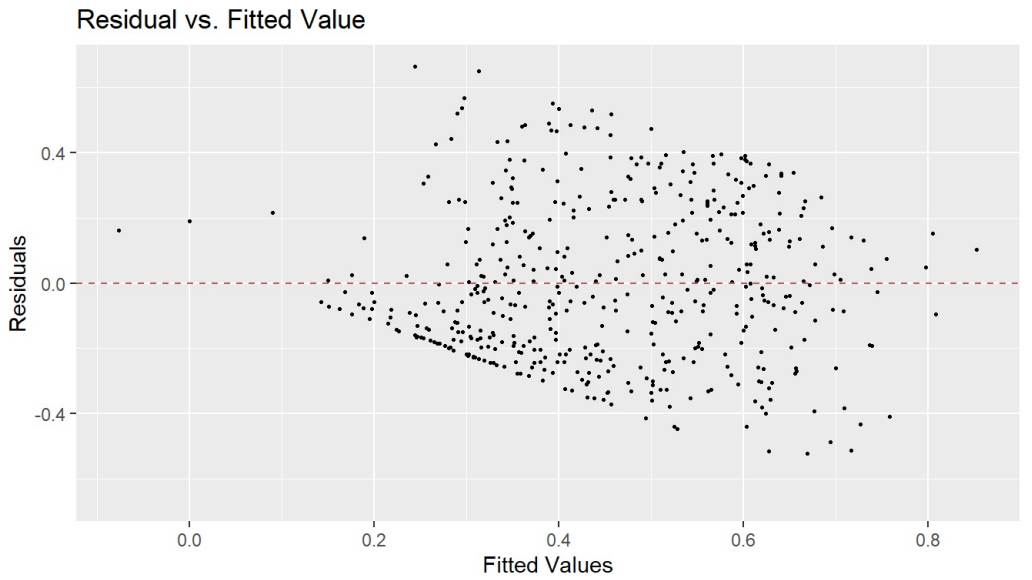
```
spp_mod %>%
  gg_diagnose(max.per.page = 1)
```

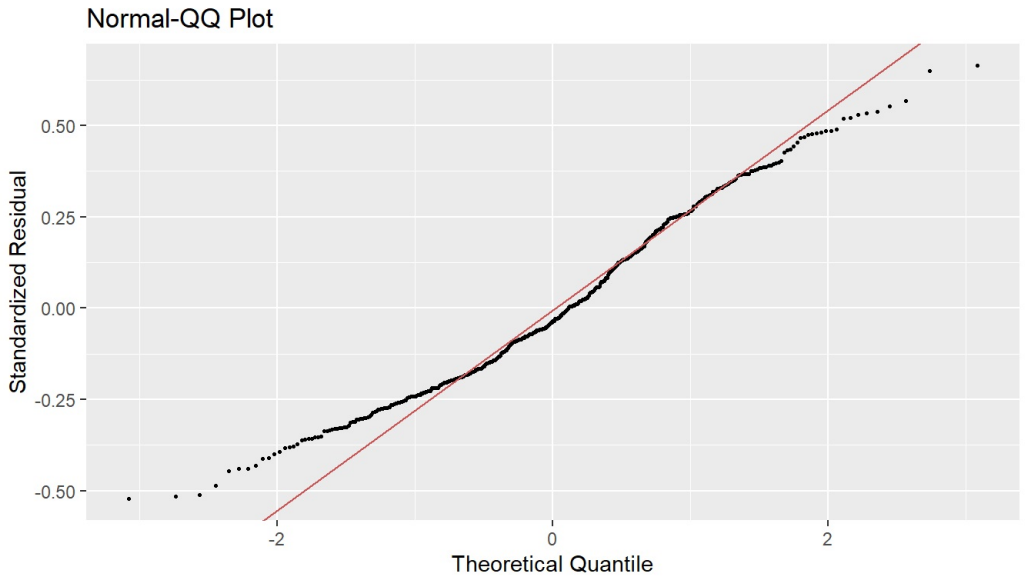

Histogram of Residuals
count
30
20
10
0
-0.50
-0.25
0.00
0.25
0.50
Residuals

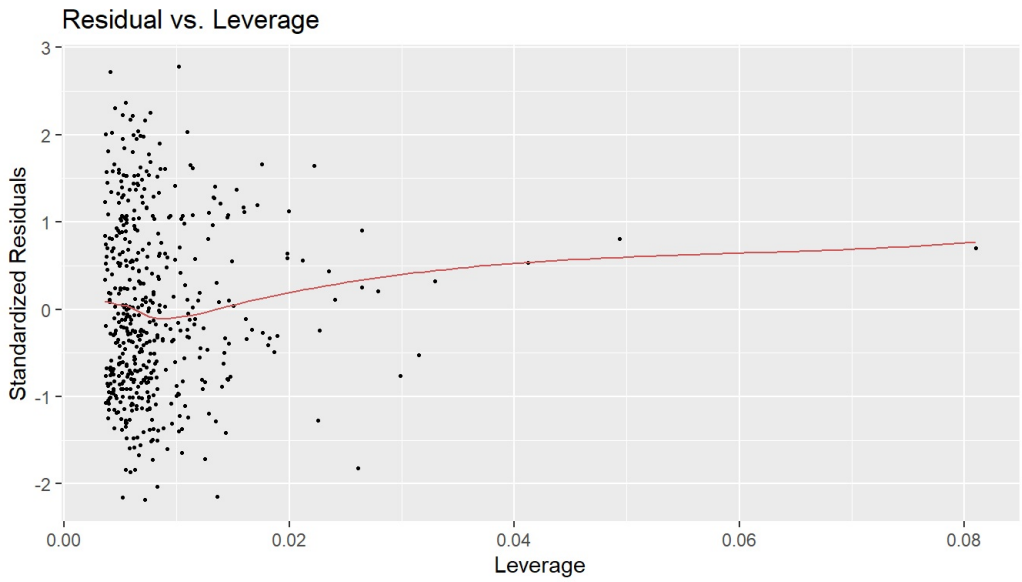

Residual vs. win_pct
residuals
0.4
0.0
-0.4
0.3
0.4
0.5
0.6
0.7
win_pct

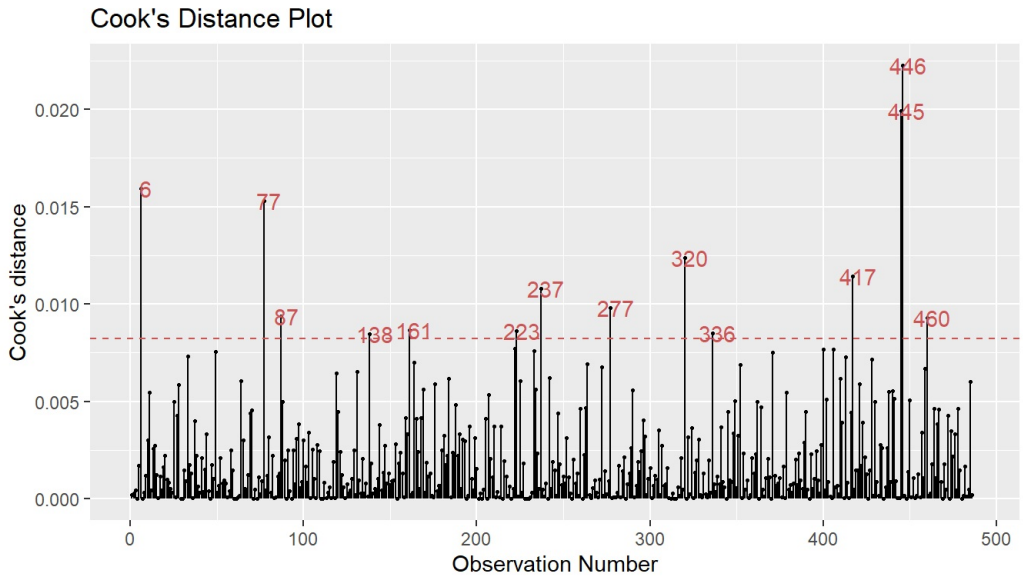

Residual vs. div_win
Residuals
0.50
0.25
0.00
-0.25
-0.50
N
Y
div_win


Residual vs. cs
residuals
0.4
0.0
-0.4
0
30
60
90
120
cs


Residual vs. Fitted Value
Residuals
0.4
0.0
-0.4
0.0
0.2
0.4
0.6
0.8
Fitted Values


Normal-QQ Plot
Standardized Residual
0.50
0.25
0.00
-0.25
-0.50
-2
0
2
Theoretical Quantile





Assumption 1: The relationship between the predictors (`win_pct` , `div_win` , `cs`) and the response variable (sqr_point_pct) is linear.

To evaluate this, a Plot the residuals against the fitted values is created. The non-random pattern in this plot indicates potential issues with the assumption of homoscedasticity (constant variance of residuals). This suggests that the variability of residuals changes with the fitted values, potentially pointing to heteroscedasticity or a missing non-linear relationship.

Assumption 2: Residuals are independent (no autocorrelation).

Since this is not time-series data, independence can continue to be assumed if data collection methods were appropriate.

Assumption 3: The variance of residuals is constant across all levels of the predictors.

This can be evaluated using the scale-location plot (standardized residuals vs. fitted values). The spread of points in this plot reinforces the concern about heteroscedasticity, as it indicates that the variance of residuals is not constant across the range of fitted values.

Assumption 4: Residuals are normally distributed.

This can be evaluated through the Q-Q plot of the residuals. While the points align reasonably well with the diagonal line, some deviations might occur at the tails, suggesting that the residuals are approximately normally distributed, though there could be slight departures from normality.

Overall Assessment: The non-random pattern in the residuals vs. fitted values plot and the spread in the scale-location plot indicate violations of the homoscedasticity assumption, while the normality assumption appears largely valid. These issues could affect the reliability of inference.

Question 1e

```
# Create a new data frame with the specified predictor values
new_data <- data.frame(
  win_pct = 0.8,
  div_win = "Y",  # 'Yes' in the dataset is encoded as "Y"
  cs = 8
)

# Predict the expected value of sqr_point_pct
predicted_value <- predict(spp_mod, newdata = new_data)

predicted_value
```

```
##         1
## 0.8770124
```

The predicted value of `sqr_point_pct` for a team with a `win_pct` of 80%, a division win (`DivWin` = Yes), and 8 Championship Series appearances is 0.87. This relatively high value indicates that teams with high winning percentages and strong postseason success are likely to perform well in terms of `sqr_point_pct` . This aligns with expectations, as better-performing teams typically achieve better outcomes in advanced metrics like `sqr_point_pct` .

Question 1f

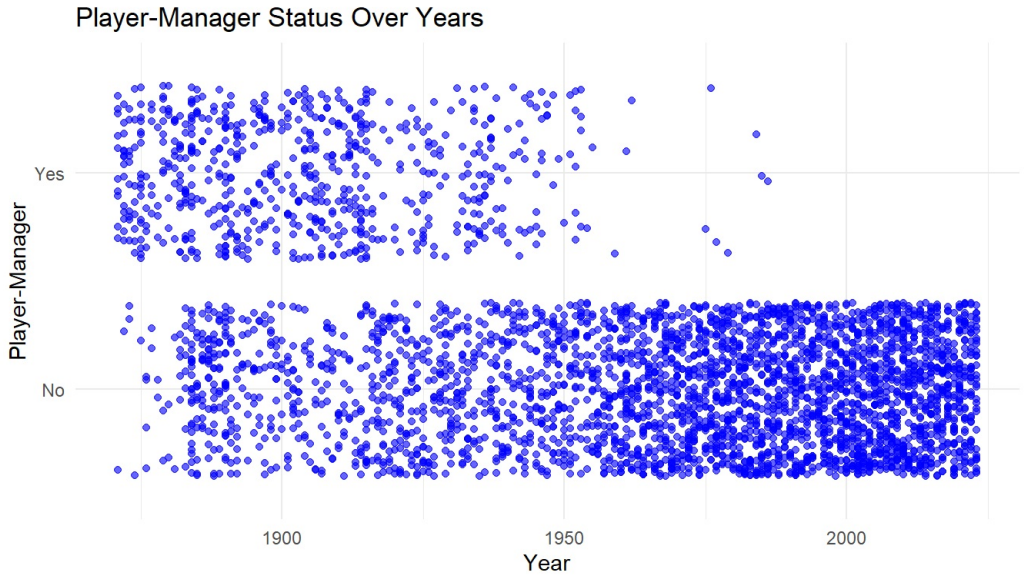
```
# Construct 95% confidence intervals
conf_intervals <- confint(spp_mod)

# Display the results
conf_intervals
```

```
##                    2.5 %       97.5 %
## (Intercept) -1.004890669 -0.450863854
## win_pct      1.313440582  2.303314302
## div_winY     0.084064590  0.188760923
## cs           0.001493985  0.003949758
```

The 95% confidence intervals confirm that all predictors in the model are statistically significant contributors to `sqr_point_pct` . The confidence interval for the intercept (-1.004 to -0.452) suggests that the baseline `sqr_point_pct` (when all predictors are zero) is significantly below zero. For `win_pct` , the interval (1.312 to 2.304) indicates a strong positive relationship, with `sqr_point_pct` increasing by approximately 1.81 for each unit increase in `win_pct` . The confidence interval for `DivWin` (0.083 to 0.190) confirms that division-winning teams see an average increase of about 0.14 in `sqr_point_pct` . Similarly, the interval for `CS` (0.001 to 0.004) highlights a small but significant positive impact of this variable.

Question 2a

```
# Plot plyr_mgr against year with vertical jitter
ggplot(data = df_managers, aes(x = year_id, y = plyr_mgr)) +
  geom_jitter(height = 0.4, width = 0, alpha = 0.6, color = "blue") +
  scale_y_discrete(labels = c("No", "Yes")) +
  labs(
    title = "Player-Manager Status Over Years",
    x = "Year",
    y = "Player-Manager"
  ) +
  theme_minimal()
```



In earlier years, a higher concentration of points corresponds to “Yes,” suggesting that player-managers were more common in the past.Over time, the density shifts toward “No,” indicating a decline in the prevalence of player-managers as the role of a manager became more specialized.

Question 2b

```
linmod5 <- glm(as.factor(plyr_mgr) ~ year_id, family = "binomial", data = df_managers)

summary(linmod5)
```

```
##
## Call:
## glm(formula = as.factor(plyr_mgr) ~ year_id, family = "binomial",
##     data = df_managers)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.79817  -0.38469  -0.18089  -0.09897   2.82264
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 88.604237   3.412071   25.97   <2e-16 ***
## year_id     -0.046611   0.001779  -26.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3442.4  on 3748  degrees of freedom
## Residual deviance: 2127.7  on 3747  degrees of freedom
## AIC: 2131.7
##
## Number of Fisher Scoring iterations: 6
```

Form of fitted Model

The estimate for the intercept is 88.6042. In the logistic regression model, this represents the log-odds of being a player-manager (`plyrMgr=1`) when the year (`yearID`) is zero. Since this scenario is outside the data’s range, it does not have a practical interpretation but serves as a baseline for calculating predictions.

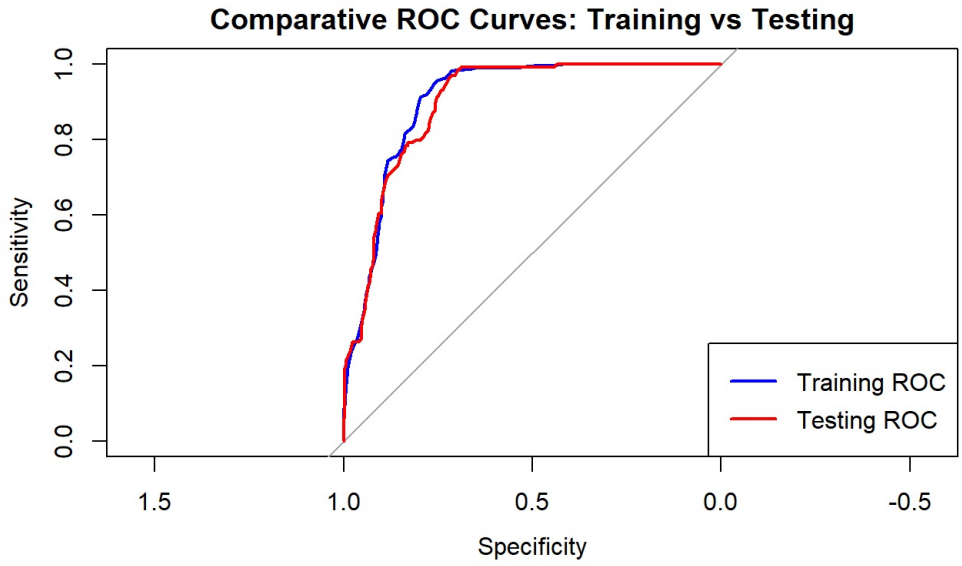
The estimate for `year_id` is -0.0466, indicating a negative association with the likelihood of being a player-manager. This suggests that the role of player-manager has become less common over time.Both the intercept and the slope (`year_id`) are statistically significant ($p<2e-16$), meaning they have a meaningful effect on the likelihood of being a player-manager.

A significant drop in deviance (from 3442.4 to 2127.7) suggests the model explains a substantial portion of the variation in the outcome.

$$\text{logit}(\Pr(\text{plyr\_mgr} = 1)) = 88.60 - 0.0466 \cdot \text{year\_id}$$

Question 2c

```
# Set seed for reproducibility
set.seed(123)

# Split data into training (80%) and testing (20%) sets
train_index <- createDataPartition(df_managers$plyr_mgr, p = 0.8, list = FALSE)
train_data <- df_managers[train_index, ]
test_data <- df_managers[-train_index, ]

table(train_data$plyr_mgr)
table(test_data$plyr_mgr)

# Fit logistic regression model on the training data
logit_train <- glm(plyr_mgr ~ year_id, family = "binomial", data = train_data)

# Predict probabilities for training and testing datasets
train_probs <- predict(logit_train, train_data, type = "response")
test_probs <- predict(logit_train, test_data, type = "response")

# Compute ROC curves
train_roc <- roc(as.numeric(train_data$plyr_mgr) - 1, train_probs)
test_roc <- roc(as.numeric(test_data$plyr_mgr) - 1, test_probs)

# Plot ROC curves
plot(train_roc, col = "blue", lwd = 2, main = "Comparative ROC Curves: Training vs Testing")
lines(test_roc, col = "red", lwd = 2)
legend("bottomright", legend = c("Training ROC", "Testing ROC"), col = c("blue", "red"), lwd = 2)
```




```
cat("Training AUC:", auc(train_roc), "\n")
cat("Testing AUC:", auc(test_roc), "\n")
```

```
##
##    N    Y
## 2484  516
##
##   N   Y
## 620 129
## Training AUC: 0.9050165
## Testing AUC: 0.8981933
```

Both the AUC values are similar and reasonably high, therefore the model generalizes well.

Question 2d

```
youden_train <- coords(train_roc, "b", best.method = "youden", transpose = TRUE)
optimal_cutoff <- youden_train[1]  # Extract the threshold directly as the first element
print(youden_train)

# Create confusion matrix for training data
train_preds <- ifelse(train_probs >= optimal_cutoff, 1, 0)
train_conf_matrix <- table(Predicted = train_preds, Actual = as.numeric(train_data$plyr_mgr) - 1)
print(train_conf_matrix)

# Create confusion matrix for testing data using the optimal cutoff
test_preds <- ifelse(test_probs >= optimal_cutoff, 1, 0)
test_conf_matrix <- table(Predicted = test_preds, Actual = as.numeric(test_data$plyr_mgr) - 1)
print(test_conf_matrix)
```

```
##   threshold specificity sensitivity
##   0.1071893   0.7552335   0.9534884
##          Actual
## Predicted    0    1
##         0 1876   24
##         1  608  492
##          Actual
## Predicted   0   1
##         0 450   6
##         1 170 123
```

The model demonstrates strong sensitivity (95.3%) at the optimal threshold of 0.107, effectively identifying most positive cases. However, its specificity (75.5%) is moderate, leading to a notable false positive rate. On the training data, the model correctly classified 1876 negatives and 492 positives, with only 24 false negatives but a significant 608 false positives. Similarly, in the testing data, it identified 450 true negatives and 123 true positives, with 6 false negatives and 170 false positives.

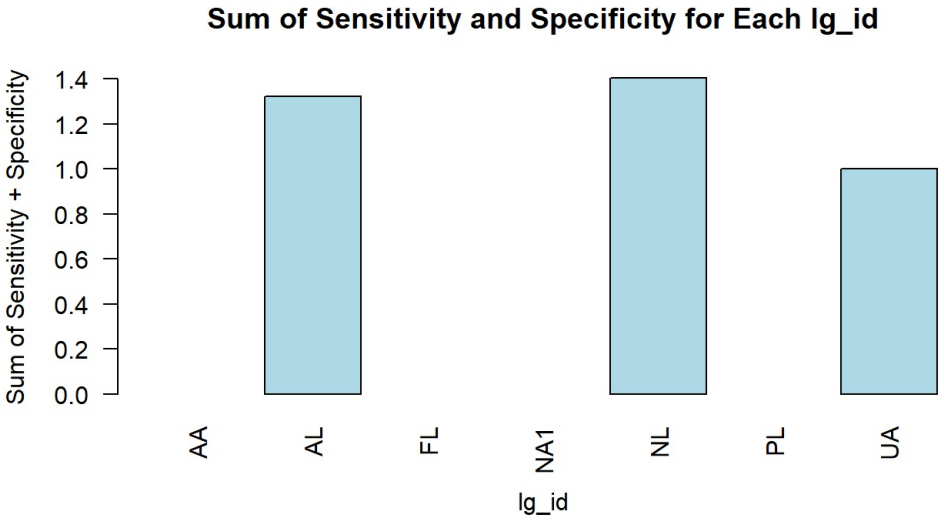
While the model generalises well between training and testing datasets, the high false positive rate reflects a trade-off caused by the low threshold.

Question 2e

```
test_preds <- ifelse(test_probs >= optimal_cutoff, 1, 0)

test_conf_matrix <- table(Predicted = test_preds, Actual = as.numeric(test_data$plyr_mgr) - 1)
print(test_conf_matrix)
```

```
##          Actual
## Predicted   0   1
##         0 450   6
##         1 170 123
```

```r
# Loop through each unique lg_id
for (lg in levels(test_data$lg_id)) {

  # Subset data for the current lg_id
  lg_subset <- test_data[test_data$lg_id == lg, ]

  # Skip if the subset is empty
  if (nrow(lg_subset) == 0) {
    next
  }

  # Predict probabilities for the current subset
  lg_subset$predicted_probs <- predict(logit_train, lg_subset, type = "response")

  # Generate binary predictions based on the optimal cutoff from part (d)
  lg_subset$predicted <- ifelse(lg_subset$predicted_probs >= optimal_cutoff, 1, 0)

  # Create confusion matrix (Predicted vs Actual)
  lg_confusion <- table(Predicted = lg_subset$predicted, Actual = as.numeric(lg_subset$plyr_mgr) - 1)

  # Print confusion matrix for the current lg_id
  cat("Confusion Matrix for lg_id:", lg, "\n")
  print(lg_confusion)
  cat("\n")
}

# Define the confusion matrices for each lg_id
conf_matrices <- list(
  AA = matrix(c(15, 8, 0, 0), nrow = 2, byrow = TRUE, dimnames = list("Predicted" = c(0, 1), "Actual" = c(0, 1)))
,
  AL = matrix(c(221, 4, 66, 34), nrow = 2, byrow = TRUE, dimnames = list("Predicted" = c(0, 1), "Actual" = c(0, 1
))),
  FL = matrix(c(0, 3, 0, 0), nrow = 2, byrow = TRUE, dimnames = list("Predicted" = c(0, 1), "Actual" = c(0, 1))),
  NA1 = matrix(c(2, 15, 0, 0), nrow = 2, byrow = TRUE, dimnames = list("Predicted" = c(0, 1), "Actual" = c(0, 1))
),
  NL = matrix(c(229, 2, 84, 59), nrow = 2, byrow = TRUE, dimnames = list("Predicted" = c(0, 1), "Actual" = c(0, 1
))),
  PL = matrix(c(0, 3, 0, 0), nrow = 2, byrow = TRUE, dimnames = list("Predicted" = c(0, 1), "Actual" = c(0, 1))),
  UA = matrix(c(3, 1, 3, 1), nrow = 2, byrow = TRUE, dimnames = list("Predicted" = c(0, 1), "Actual" = c(0, 1)))
)

# Function to compute sensitivity and specificity and sum them
compute_sensitivity_specificity_sum <- function(conf_matrix) {
  tp <- conf_matrix[2, 2]  # True positives
  tn <- conf_matrix[1, 1]  # True negatives
  fp <- conf_matrix[1, 2]  # False positives
  fn <- conf_matrix[2, 1]  # False negatives

  sensitivity <- tp / (tp + fn)  # Sensitivity = TP / (TP + FN)
  specificity <- tn / (tn + fp)  # Specificity = TN / (TN + FP)

  return(sensitivity + specificity)  # Return the sum
}

# Calculate and print the sum for each lg_id
sensitivity_specificity_sums <- sapply(conf_matrices, compute_sensitivity_specificity_sum)
sensitivity_specificity_sums

# Plot the sums as a bar chart
barplot(sensitivity_specificity_sums,
        names.arg = names(sensitivity_specificity_sums),
        col = "lightblue",
        main = "Sum of Sensitivity and Specificity for Each lg_id",
        ylab = "Sum of Sensitivity + Specificity",
        xlab = "lg_id",
        las = 2)
```

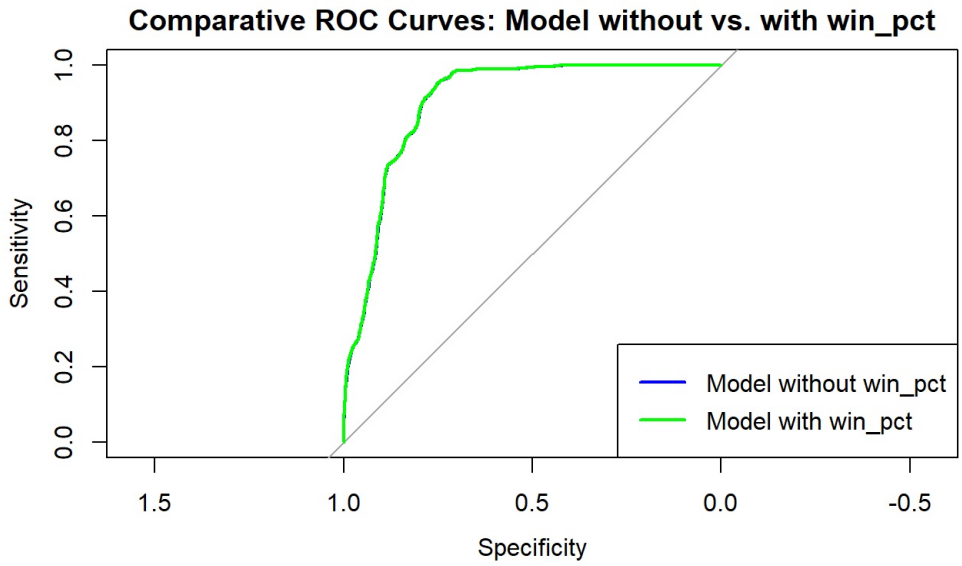



```
## Confusion Matrix for lg_id: AA
##          Actual
## Predicted  0  1
##         1 15  8
##
## Confusion Matrix for lg_id: AL
##          Actual
## Predicted   0   1
##         0 221   4
##         1  66  34
##
## Confusion Matrix for lg_id: FL
##          Actual
## Predicted 1
##         1 3
##
## Confusion Matrix for lg_id: NA
##          Actual
## Predicted  0  1
##         1  2 15
##
## Confusion Matrix for lg_id: NL
##          Actual
## Predicted   0   1
##         0 229   2
##         1  84  59
##
## Confusion Matrix for lg_id: PL
##          Actual
## Predicted 1
##         1 3
##
## Confusion Matrix for lg_id: UA
##          Actual
## Predicted 0 1
##         1 3 1
##
##       AA       AL       FL      NA1       NL       PL       UA
##      NaN 1.322222      NaN      NaN 1.403929      NaN 1.000000
```

The bar chart reveals that the AL, NL, and UA groups have positive sums of sensitivity and specificity, with values of 1.32, 1.40, and 1.00, respectively. The AA, FL, NA1, and PL groups have NaN values, indicating missing or undefined data, which suggests that these groups did not yield meaningful results for sensitivity and specificity. The high values for AL and NL suggest better model performance in these group in both identifying true positives and negatives.

Question 2f

```
# Fit a new logistic regression model with the additional 'win_pct' variable
linmod6 <- glm(as.factor(plyr_mgr) ~ year_id + win_pct, family = "binomial", data = df_managers)

# Summary of the new model
summary(linmod6)

# Compare the AIC of both models
cat("AIC of the model without win_pct (linmod5):", AIC(linmod5), "\n")
cat("AIC of the model with win_pct (linmod6):", AIC(linmod6), "\n")

# Compare the models using ROC curves and AUC (optional)
train_probs_linmod6 <- predict(linmod6, df_managers, type = "response")
train_roc_linmod6 <- roc(as.numeric(df_managers$plyr_mgr) - 1, train_probs_linmod6)

# Plot the ROC curves for both models
train_probs_linmod5 <- predict(linmod5, df_managers, type = "response")
train_roc_linmod5 <- roc(as.numeric(df_managers$plyr_mgr) - 1, train_probs_linmod5)

plot(train_roc_linmod5, col = "blue", lwd = 2, main = "Comparative ROC Curves: Model without vs. with win_pct")
lines(train_roc_linmod6, col = "green", lwd = 2)
legend("bottomright", legend = c("Model without win_pct", "Model with win_pct"), col = c("blue", "green"), lwd =
2)
```




```
# Print AUC values for comparison
cat("without win_pct:", auc(train_roc_linmod5), "\n")
cat("with win_pct:", auc(train_roc_linmod6), "\n")
```

```
##
## Call:
## glm(formula = as.factor(plyr_mgr) ~ year_id + win_pct, family = "binomial",
##     data = df_managers)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.79833  -0.38456  -0.18093  -0.09893   2.82252
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 88.615550   3.419632  25.914   <2e-16 ***
## year_id     -0.046622   0.001793 -26.009   <2e-16 ***
## win_pct      0.020262   0.395959   0.051    0.959
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3442.4  on 3748  degrees of freedom
## Residual deviance: 2127.7  on 3746  degrees of freedom
## AIC: 2133.7
##
## Number of Fisher Scoring iterations: 6
##
## AIC of the model without win_pct (linmod5): 2131.66
## AIC of the model with win_pct (linmod6): 2133.658
## without win_pct: 0.9037983
## with win_pct: 0.9037703
```

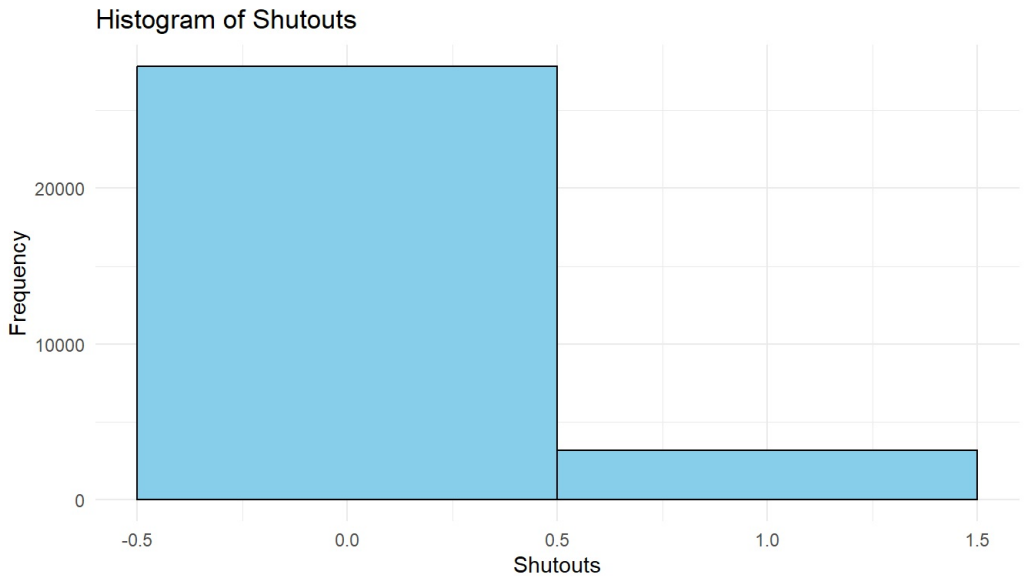
The model with `year_id` and `win_pct` shows that while `year_id` is highly significant ($p < 0.001$) with a negative coefficient (-0.046622), indicating a decreasing likelihood of becoming a manager over time, `win_pct` is not significant ($p = 0.959$) and has no meaningful impact. Adding `win_pct` slightly increases the AIC (2133.66 vs. 2131.66) and has no effect on the model's discriminatory power, with nearly identical AUC values (0.9038 vs. 0.9037). Thus, the simpler model without win_pct is preferred for its better fit and parsimony.

Question 3a

```
df_pitchers <- Pitching[Pitching$IPouts > 0, ]

#Add the 'innings' variable, which is IPouts/3
df_pitchers$innings <- df_pitchers$IPouts / 3

df_pitchers <- merge(df_pitchers, People[, c("playerID", "weight", "height", "throws")], by = "playerID")

# Remove incomplete cases
df_pitchers <- na.omit(df_pitchers)
```

Question 3b

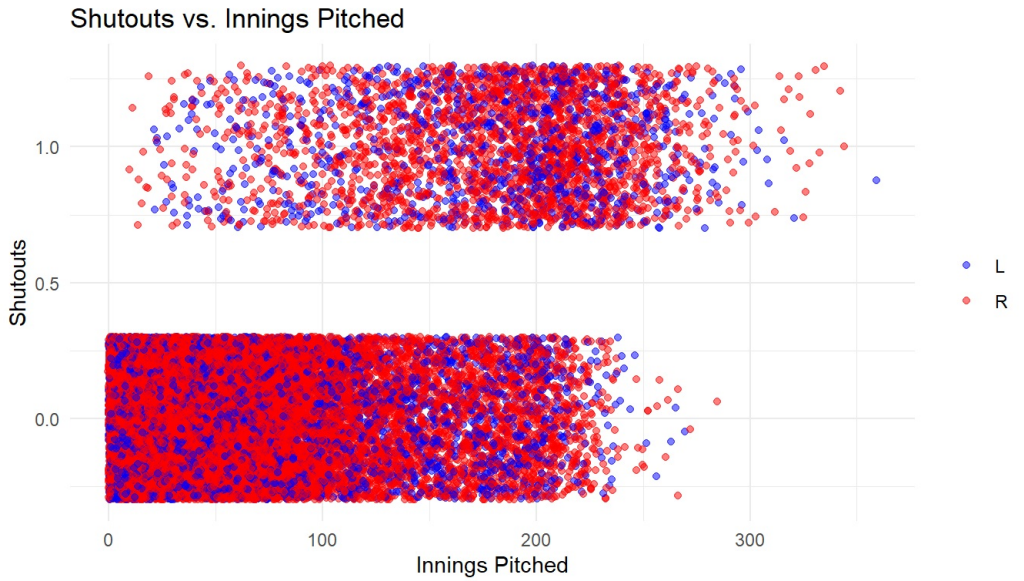
```
# Create a variable for shutouts (assuming 'shutouts' is not available directly)
df_pitchers$shutouts <- ifelse(df_pitchers$SHO > 0, 1, 0)

#Plot a histogram of shutouts
library(ggplot2)

ggplot(df_pitchers, aes(x = shutouts)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
  labs(title = "Histogram of Shutouts",
       x = "Shutouts",
       y = "Frequency") +
  theme_minimal()
```



A Poisson model is suitable for the number of shutouts because shutouts are relatively rare occurrences in baseball. Most pitchers do not record shutouts frequently. Also, We are counting the number of shutouts for each pitcher, which is a type of count data. The occurrence of shutouts can be assumed to be independent events for each pitcher over a season or career.Also, Poisson distributions often assume that the mean and variance of the count data are approximately equal, which is generally the case for rare events like shutouts.

Question 3c

```
#glimpse(df_pitchers)
levels(df_pitchers$throws)

# Filter for left-handed and right-handed pitchers
df_filtered <- df_pitchers[df_pitchers$throws %in% c("L", "R"), ]

# Create the plot
ggplot(df_filtered, aes(x = innings, y = shutouts, color = throws)) +
  geom_jitter(width = 0.7, height = 0.3, alpha = 0.5) +  # Jitter the data vertically
  labs(title = "Shutouts vs. Innings Pitched",
       x = "Innings Pitched",
       y = "Shutouts") +
  scale_color_manual(values = c("blue", "red")) +  # Color for left and right handed pitchers
  theme_minimal() +
  theme(legend.title = element_blank())
```




```
## [1] "B" "L" "R" "S"
```

we can see from the graph that typically, pitchers who throw more innings may have a higher chance of earning a shutout, but the relationship seems to be influenced by other factors (e.g., the quality of the opposing team or the pitcher's performance). The points are spread similarly across the two colors, this suggests that handedness does not have a strong effect on the occurrence of shutouts.

Question 3d

```
# Fit the Poisson regression model
poisson_mod1 <- glm(shutouts ~ innings + weight + height + throws,
                    family = "poisson", data = df_pitchers)

# Summarize the model
summary(poisson_mod1)

anova(poisson_mod1, test = "Chisq")
```

```
##
## Call:
## glm(formula = shutouts ~ innings + weight + height + throws,
##     family = "poisson", data = df_pitchers)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.0783  -0.2899  -0.2067  -0.1685   2.6027
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.234e+00  6.627e-01 -10.917  < 2e-16 ***
## innings      1.773e-02  2.256e-04  78.590  < 2e-16 ***
## weight      -6.525e-03  1.010e-03  -6.461 1.04e-10 ***
## height       5.680e-02  1.008e-02   5.634 1.76e-08 ***
## throwsR     -7.877e-02  3.872e-02  -2.034   0.0419 *
## throwsS     -9.162e+00  1.908e+02  -0.048   0.9617
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 14474.4  on 31010  degrees of freedom
## Residual deviance:  7434.4  on 31005  degrees of freedom
## AIC: 13798
##
## Number of Fisher Scoring iterations: 11
```

NULL
innings
weight
height
throws

5 rows | 1-1 of 6 columns

The intercept is highly significant (< 2e-16), indicating that when all predictors are zero (a hypothetical situation), the log of the expected shutouts is -7.234.The p-value is extremely small for both the weight, the height, and the innings, indicating that they are highly significant in predicting the number of shutouts. The predictor `Throws` is significant (0.0419), though less so than innings, weight, and height.

Question 3e

```
# Randomize the teamID column
set.seed(123)  # For reproducibility
df_pitchers$random_teamID <- sample(df_pitchers$teamID, size = nrow(df_pitchers), replace = FALSE)

# Ensure random_teamID is a factor
df_pitchers$random_teamID <- as.factor(df_pitchers$random_teamID)

# Fit the Poisson regression model with randomized teamID as a fixed effect
poisson_mod2 <- glm(
  shutouts ~ innings + weight + height + throws + random_teamID,
  family = "poisson",
  data = df_pitchers
)

# Summarize the model
summary(poisson_mod2)

# Extract the coefficients and round them to 2 significant figures
coefficients <- round(coef(poisson_mod2), 2)
print(coefficients)
```

```
##
## Call:
## glm(formula = shutouts ~ innings + weight + height + throws +
##     random_teamID, family = "poisson", data = df_pitchers)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8150  -0.2889  -0.2062  -0.1677   2.5903
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -7.053e+00  7.047e-01 -10.008  < 2e-16 ***
## innings           1.778e-02  2.272e-04  78.238  < 2e-16 ***
## weight           -6.474e-03  1.013e-03  -6.394 1.62e-10 ***
## height            5.630e-02  1.012e-02   5.562 2.67e-08 ***
## throwsR          -8.019e-02  3.887e-02  -2.063   0.0391 *
## throwsS          -9.138e+00  1.912e+02  -0.048   0.9619
## random_teamIDARI -2.155e-01  2.501e-01  -0.862   0.3889
## random_teamIDATL -4.870e-02  2.355e-01  -0.207   0.8362
## random_teamIDBAL -1.592e-01  2.386e-01  -0.667   0.5045
## random_teamIDBOS -7.534e-02  2.380e-01  -0.317   0.7516
## random_teamIDCAL -1.778e-01  2.685e-01  -0.662   0.5078
## random_teamIDCHA -2.037e-01  2.397e-01  -0.850   0.3955
## random_teamIDCHN -2.096e-01  2.403e-01  -0.872   0.3830
## random_teamIDCIN -2.032e-01  2.378e-01  -0.854   0.3929
## random_teamIDCLE -9.898e-02  2.377e-01  -0.416   0.6771
## random_teamIDCOL -9.386e-02  2.481e-01  -0.378   0.7052
## random_teamIDDET -8.871e-02  2.359e-01  -0.376   0.7069
## random_teamIDFLO -1.084e-02  2.708e-01  -0.040   0.9681
## random_teamIDHOU -3.092e-01  2.419e-01  -1.278   0.2012
## random_teamIDKCA -2.931e-01  2.372e-01  -1.236   0.2166
## random_teamIDLAA -1.631e-01  2.609e-01  -0.625   0.5318
## random_teamIDLAN -1.495e-01  2.382e-01  -0.628   0.5303
## random_teamIDMIA -2.204e-02  2.776e-01  -0.079   0.9367
## random_teamIDMIL -6.895e-02  2.466e-01  -0.280   0.7798
## random_teamIDMIN -2.177e-01  2.412e-01  -0.902   0.3668
## random_teamIDML4 -1.193e-01  2.721e-01  -0.438   0.6611
## random_teamIDMON -2.206e-01  2.493e-01  -0.885   0.3763
## random_teamIDNYA -5.369e-02  2.350e-01  -0.228   0.8193
## random_teamIDNYN -1.691e-01  2.382e-01  -0.710   0.4777
## random_teamIDOAK -1.237e-01  2.380e-01  -0.520   0.6032
## random_teamIDPHI -1.265e-01  2.348e-01  -0.539   0.5901
## random_teamIDPIT -3.087e-01  2.387e-01  -1.293   0.1960
## random_teamIDSDN -2.407e-01  2.401e-01  -1.002   0.3161
## random_teamIDSEA -1.815e-01  2.386e-01  -0.761   0.4469
## random_teamIDSFN -1.434e-01  2.410e-01  -0.595   0.5517
## random_teamIDSLN -1.605e-01  2.375e-01  -0.676   0.4991
## random_teamIDTBA -1.686e-01  2.511e-01  -0.672   0.5019
## random_teamIDTEX -1.980e-01  2.364e-01  -0.838   0.4021
## random_teamIDTOR -1.514e-01  2.375e-01  -0.637   0.5240
## random_teamIDWAS -1.319e-01  2.626e-01  -0.502   0.6154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 14474.4  on 31010  degrees of freedom
## Residual deviance:  7417.2  on 30971  degrees of freedom
## AIC: 13849
##
## Number of Fisher Scoring iterations: 11
##
##      (Intercept)          innings           weight           height
##            -7.05             0.02            -0.01             0.06
##           throwsR          throwsS random_teamIDARI random_teamIDATL
##            -0.08            -9.14            -0.22            -0.05
## random_teamIDBAL random_teamIDBOS random_teamIDCAL random_teamIDCHA
##            -0.16            -0.08            -0.18            -0.20
## random_teamIDCHN random_teamIDCIN random_teamIDCLE random_teamIDCOL
##            -0.21            -0.20            -0.10            -0.09
## random_teamIDDET random_teamIDFLO random_teamIDHOU random_teamIDKCA
##            -0.09            -0.01            -0.31            -0.29
## random_teamIDLAA random_teamIDLAN random_teamIDMIA random_teamIDMIL
##            -0.16            -0.15            -0.02            -0.07
## random_teamIDMIN random_teamIDML4 random_teamIDMON random_teamIDNYA
##            -0.22            -0.12            -0.22            -0.05
## random_teamIDNYN random_teamIDOAK random_teamIDPHI random_teamIDPIT
##            -0.17            -0.12            -0.13            -0.31
```

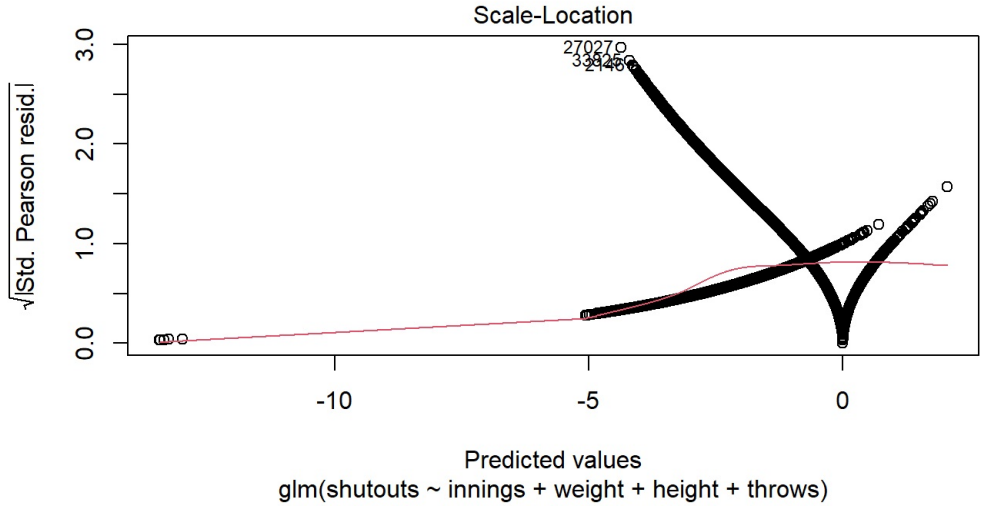
```
## random_teamIDSDN random_teamIDSEA random_teamIDSFN random_teamIDSLN
##            -0.24            -0.18            -0.14            -0.16
## random_teamIDTBA random_teamIDTEX random_teamIDTOR random_teamIDWAS
##            -0.17            -0.20            -0.15            -0.13
```

$$\log(E[\text{shutouts}]) = -7.05 + 0.02 \cdot \text{innings} - 0.01 \cdot \text{weight} + 0.06 \cdot \text{height} - 0.08 \cdot \text{throwsR} - 9.14 \cdot \text{throwsS} + \sum_{i=1}^{n} \gamma_i \cdot \text{random\_teamID}_i$$

From the model results, `teamID` does not appear to be a significant predictor of the outcome variable, as indicated by the p-values for the random effect terms associated with each team. Most of the p-values for the `random_teamID` coefficients are greater than 0.05, which suggests that there is no strong evidence to conclude that `teamID` has a significant impact on the shutouts. The random effect for teamID likely does not explain much of the variation in the outcome beyond what is already captured by the other predictors such as innings, weight, height, and throws. Therefore, `teamID` does not seem to be an important predictor in this model.

Question 3f

```
# Produce the Scale-Location plot for poisson_mod1
plot(poisson_mod1, which = 3)
```



The Scale-Location plot reveals a funnel-shaped pattern, indicating potential heteroscedasticity, where the residuals' spread increases as the predicted values become more extreme. This suggests that the variance in the data is not constant, which could be a sign of overdispersion in the Poisson model. Additionally, there are a few outliers with extreme residuals, which may be influencing the model's fit. These points warrant further investigation to determine if they represent data entry errors or significant extreme cases. To address these issues, it may be necessary to test for overdispersion and consider alternative models like Negative Binomial regression.

Question 3g

```
# 1. Extract the coefficients from the model
coefs <- summary(poisson_mod1)$coefficients[, 1]

print(coefs)
# 2. Calculate the relative rate for left-handed pitchers (throwsL)
throwsL_coef <- coefs["throwsS"]  # Coefficient for left-handed pitchers
print(throwsL_coef)
relative_rate_throwsL <- exp(throwsL_coef)
cat("Relative rate of shutouts for left-handed pitchers compared to right-handed pitchers:", relative_rate_throws
L, "\n")

# 3. Calculate the relative rate for height (taller players)
height_coef <- coefs["height"]  # Coefficient for height
relative_rate_height <- exp(height_coef)
cat("Relative rate of shutouts for taller players compared to shorter players:", relative_rate_height, "\n")

# 4. Calculate the relative rate for weight (heavier players)
weight_coef <- coefs["weight"]  # Coefficient for weight
relative_rate_weight <- exp(weight_coef)
cat("Relative rate of shutouts for heavier players compared to lighter players:", relative_rate_weight, "\n")
```

```
##  (Intercept)      innings       weight       height      throwsR      throwsS
## -7.234289195  0.017733250 -0.006524699  0.056799416 -0.078771044 -9.161875686
##   throwsS
## -9.161876
## Relative rate of shutouts for left-handed pitchers compared to right-handed pitchers: 0.0001049658
## Relative rate of shutouts for taller players compared to shorter players: 1.058443
## Relative rate of shutouts for heavier players compared to lighter players: 0.9934965
```

Left-handed pitchers are expected to pitch fewer shutouts than right-handed pitchers. Taller players are expected to pitch more shutouts than shorter players. Heavier players are expected to pitch fewer shutouts than lighter players

A very strong negative relationship between being left-handed and pitching shutouts could be due to less frequent opportunities for left-handed pitchers to start games or facing more challenges in terms of matchups against certain teams. The positive relationship between height and shutouts may be attributed to factors like better reach, stride length, which could help them perform better on the mound. Although heavier players are expected to pitch fewer shutouts than lighter players, the small difference in the relative rate (0.9935) suggests that weight has a minimal impact on pitching performance. Other factors like skill, stamina, and technique likely play a more significant role in shutout performance than body weight alone.