# NFL Assignment

2026-02-14

## ~~~ Data Exploration ~~~

## 1. Find the highest and lowest salary in the dataset.

```r
# Highest and lowest salary

#base R
max(nfl$Salary, na.rm = TRUE)
```

```
## [1] 20100000
```

```r
min(nfl$Salary, na.rm = TRUE)
```

```
## [1] 48333
```

```r
#tidyverse way
nfl %>%
    summarise( max_sal = max( Salary, na.rm = TRUE ),
               min_sal = min( Salary, na.rm = TRUE ) )
```

```
## # A tibble: 1 x 2
##    max_sal min_sal
##      <dbl>   <dbl>
## 1 20100000   48333
```

```r
# Answer: The highest salary is $20,100,000 and the lowest salary is $48,333.
```

## 2. Print the college and state for the five best paid players of the San Francisco 49ers

```r
nfl %>%
    filter( Team == "SF" )  %>%
    arrange( desc( Salary ) ) %>%
    select( "Player Name", College, State ) %>%
    head(5)
```

```
## # A tibble: 5 x 3
##   `Player Name`   College      State
##   <chr>           <chr>        <chr>
## 1 Willis, Patrick Mississippi  Tennessee
## 2 Rogers, Carlos  Auburn       Georgia
## 3 Davis, Vernon   Maryland     DC
## 4 Bowman, NaVorro Penn State   Maryland
## 5 Gore, Frank     Miami (Fla.) Florida
```

## 3. Calculate the mean salary and number of players in the dataset for each position

Arrange your results from highest to lowest mean salary.

Hint: You might want some of the following code. You should also use the `group_by() + summarise()` combo. Also use `summarise( n = n() )` to make a count of the number of records.

```r
# Create a cleaned dataframe with NAs removed from Salary column

clean.dat <- nfl %>%
    filter( !is.na( Salary ) )

clean.dat %>%
    group_by( Position ) %>%
    summarise( mean_salary = mean( Salary ), count = n() ) %>%
    arrange( desc( mean_salary ) )
```

```
## # A tibble: 16 x 3
##    Position mean_salary count
##    <chr>          <dbl> <int>
##  1 QB          4145334.    84
##  2 DE          2767078.   142
##  3 DL          2366991.   102
##  4 WR          2238717.   192
##  5 OL          2223450.   243
##  6 NT          2107992.    21
##  7 C           2107085.    51
##  8 CB          2029060.   176
##  9 LB          1917398.   232
## 10 S           1843162.   142
## 11 K           1719945     30
## 12 RB          1682565.   136
## 13 TE          1658034.   113
## 14 FB          1437786.    21
## 15 P           1357487.    33
## 16 LS          1091954.    29
```

## 4. Determine if taller players get paid more

Use linear regression to answer this question. For an overview of linear regression in R, see the course video and materials posted on Canvas.

Sometimes measures such as salary have such a skew that the outliers cause real problems. For doing this question, you might first try log transforming Salary via a mutate command to make `logSal = log( Salary )`, and then fitting your model on that space.

```r
# Add logSal variable and clean up a data point with a typo

clean.dat <- clean.dat %>%
    mutate( logSal = log( Salary), Height = if_else(Height == "6;2/;", "6'2\"", Height ) )


model <- lm( logSal ~ Height, data = clean.dat )
summary( model )
```

```
##
## Call:
## lm(formula = logSal ~ Height, data = clean.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1080 -0.7846 -0.3916  0.7193  2.8404
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.93006    0.09001 154.763   <2e-16 ***
## Height5'11"  0.03449    0.12176   0.283   0.7770
## Height5'5"  -0.94696    0.97775  -0.969   0.3329
## Height5'6"  -0.67147    0.69430  -0.967   0.3336
## Height5'7"  -0.19718    0.40753  -0.484   0.6286
## Height5'8"  -0.06584    0.24081  -0.273   0.7846
## Height5'9"  -0.09343    0.16450  -0.568   0.5701
## Height6'0"   0.04080    0.11549   0.353   0.7239
## Height6'1"  -0.03622    0.11185  -0.324   0.7461
## Height6'2"   0.13630    0.11106   1.227   0.2199
## Height6'3"   0.11923    0.10878   1.096   0.2732
## Height6'4"   0.15352    0.11131   1.379   0.1680
## Height6'5"   0.16856    0.11738   1.436   0.1512
## Height6'6"   0.22992    0.13566   1.695   0.0903 .
## Height6'7"   0.40544    0.18556   2.185   0.0290 *
## Height6'8"   0.23687    0.24081   0.984   0.3254
## Height6'9"  -0.10281    0.69430  -0.148   0.8823
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9736 on 1730 degrees of freedom
## Multiple R-squared:  0.01122,    Adjusted R-squared:  0.002074
## F-statistic: 1.227 on 16 and 1730 DF,  p-value: 0.2389
```

```r
exp( coef( model ) )
```

```
##  (Intercept)  Height5'11"   Height5'5"   Height5'6"   Height5'7"   Height5'8"
## 1.121370e+06 1.035095e+00 3.879185e-01 5.109589e-01 8.210410e-01 9.362762e-01
##   Height5'9"   Height6'0"   Height6'1"   Height6'2"   Height6'3"   Height6'4"
## 9.107997e-01 1.041644e+00 9.644282e-01 1.146020e+00 1.126626e+00 1.165932e+00
##   Height6'5"   Height6'6"   Height6'7"   Height6'8"   Height6'9"
## 1.183594e+00 1.258493e+00 1.499966e+00 1.267279e+00 9.022942e-01
```

Once you do, you can use `exp( coef )`, where `coef` is the coefficient from your regression, to get an interpretation of your coefficient as percent change in salary given a change in height.

Notes on our model: When fitting a model to understand whether height is associated with salary, it's usually better to treat both salary and age as continuous numeric variables. In the current model and plot, height is treated as a categorical variable, which makes the interpretation of the height–salary relationship more complicated. The plot doesn't communicate much, as the categorical variables were sorted into alphabetical order, not according to their magnitude.

With the model using categorical variables, we have the height 5'10 set as the intercept, meaning that we estimate that players with that height will make on average US$1,121,370, while players who are Height 6'2" and above will have higher salaries than the shorter players. However, none of these coefficients are statistically significant, apart from the players measuring 6'7" (t(d = 1730) = 2.18, p=0.029), who will on average make 1.5 times more than the players measuring 5'10.

## 5. Improve the linear model

What do you think about the results in (4)? If you found differences, can you explain them by accounting for player position in your model? In other words, what happens when you include position in your model?

```r
# Create a new model that incorporates both Position and Height

model <- lm( logSal ~ Height + Position, data = clean.dat )
summary( model )
```

```
##
## Call:
## lm(formula = logSal ~ Height + Position, data = clean.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1082 -0.7643 -0.3894  0.6751  2.6602
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.070672   0.179029  78.594   <2e-16 ***
## Height5'11"  0.027514   0.122272   0.225   0.8220
## Height5'5"  -0.974441   0.974187  -1.000   0.3173
## Height5'6"  -0.581247   0.693500  -0.838   0.4021
## Height5'7"  -0.146197   0.407617  -0.359   0.7199
## Height5'8"  -0.033820   0.241856  -0.140   0.8888
## Height5'9"  -0.086931   0.163919  -0.530   0.5960
## Height6'0"   0.026096   0.116781   0.223   0.8232
## Height6'1"  -0.048912   0.118509  -0.413   0.6799
## Height6'2"   0.082874   0.123366   0.672   0.5018
## Height6'3"   0.032939   0.128702   0.256   0.7980
## Height6'4"   0.074294   0.134929   0.551   0.5820
## Height6'5"   0.132291   0.144422   0.916   0.3598
## Height6'6"   0.173253   0.162709   1.065   0.2871
## Height6'7"   0.383756   0.207155   1.853   0.0641 .
## Height6'8"   0.150272   0.258332   0.582   0.5608
## Height6'9"  -0.138755   0.698388  -0.199   0.8425
## PositionCB  -0.094586   0.169102  -0.559   0.5760
## PositionDE   0.091281   0.158932   0.574   0.5658
## PositionDL   0.012462   0.166404   0.075   0.9403
## PositionFB  -0.309252   0.256521  -1.206   0.2282
## PositionK    0.013867   0.231996   0.060   0.9523
## PositionLB  -0.157652   0.153107  -1.030   0.3033
## PositionLS  -0.436833   0.226532  -1.928   0.0540 .
## PositionNT   0.003698   0.251288   0.015   0.9883
## PositionOL  -0.104671   0.153568  -0.682   0.4956
## PositionP   -0.263102   0.218909  -1.202   0.2296
## PositionQB   0.311460   0.172257   1.808   0.0708 .
## PositionRB  -0.230830   0.176630  -1.307   0.1914
## PositionS   -0.127643   0.170407  -0.749   0.4539
## PositionTE  -0.342061   0.166029  -2.060   0.0395 *
## PositionWR  -0.113130   0.159438  -0.710   0.4781
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9674 on 1715 degrees of freedom
## Multiple R-squared:  0.03225,    Adjusted R-squared:  0.01476
## F-statistic: 1.844 on 31 and 1715 DF,  p-value: 0.003247
```

```
exp( coef( model ) )
```

```
##  (Intercept)  Height5'11"   Height5'5"   Height5'6"   Height5'7"   Height5'8"
## 1.290670e+06 1.027896e+00 3.774033e-01 5.592005e-01 8.639879e-01 9.667459e-01
##   Height5'9"   Height6'0"   Height6'1"   Height6'2"   Height6'3"   Height6'4"
## 9.167405e-01 1.026439e+00 9.522651e-01 1.086405e+00 1.033487e+00 1.077124e+00
##   Height6'5"   Height6'6"   Height6'7"   Height6'8"   Height6'9"   PositionCB
## 1.141440e+00 1.189166e+00 1.467788e+00 1.162150e+00 8.704416e-01 9.097494e-01
##   PositionDE   PositionDL   PositionFB    PositionK   PositionLB   PositionLS
## 1.095577e+00 1.012540e+00 7.339959e-01 1.013964e+00 8.541467e-01 6.460791e-01
##   PositionNT   PositionOL    PositionP   PositionQB   PositionRB    PositionS
## 1.003705e+00 9.006210e-01 7.686632e-01 1.365417e+00 7.938747e-01 8.801671e-01
##   PositionTE   PositionWR
## 7.103046e-01 8.930348e-01
```

After adding Position as a covariate, we're left with no categories of height having a statistical significance, which means that according to this model, when controlling for position, no categories of height are associated with higher salaries (in terms of statistical significance). The only position that shows a somewhat statistically significant relationship with log salary is tight end (TE). R^2 increased from 0.002 to 0.015, in comparison with the previous model. In other words, this model has a slightly better fit than the one with just height. However, R^2 is still very small.

## 6. Print the mean weight and height for the football players, calculated for each position

You are going to have to get height cleaned up. You might try `separate()` (see R for DS) to make feet and inches, and then do some math via `mutate()`. You could also use `substring()`. In any case, you might find `parse_number()` a useful function. Note that one foot (abbreviated as ’) is equal to 12 inches (abbreviated to “). Try asking a Large Language Model for help with this one.

```
# Convert height into total inches

clean.dat <- clean.dat %>%
    separate( Height, into = c("feet", "inches"), sep = "'" ) %>%
    mutate( feet = parse_number( feet ),
            inches = parse_number( inches ),
            total_inches = feet * 12 + inches )

clean.dat %>%
    group_by( Position ) %>%
    summarise( mean_weight = mean( Weight, na.rm = TRUE ),
               mean_height = mean( total_inches, na.rm = TRUE ) )
```

```
## # A tibble: 16 x 3
##    Position mean_weight mean_height
##    <chr>          <dbl>       <dbl>
##  1 C               304.        75.4
##  2 CB              193.        71.4
##  3 DE              279.        76.0
##  4 DL              312.        75.0
##  5 FB              247.        72.5
##  6 K               202.        71.7
##  7 LB              245.        73.9
##  8 LS              246.        74.6
##  9 NT              322.        74.8
## 10 OL              315.        76.9
## 11 P               215.        73.8
## 12 QB              224.        75.2
## 13 RB              216.        70.6
## 14 S               207.        72.0
## 15 TE              256.        76.5
## 16 WR              201.        72.6
```

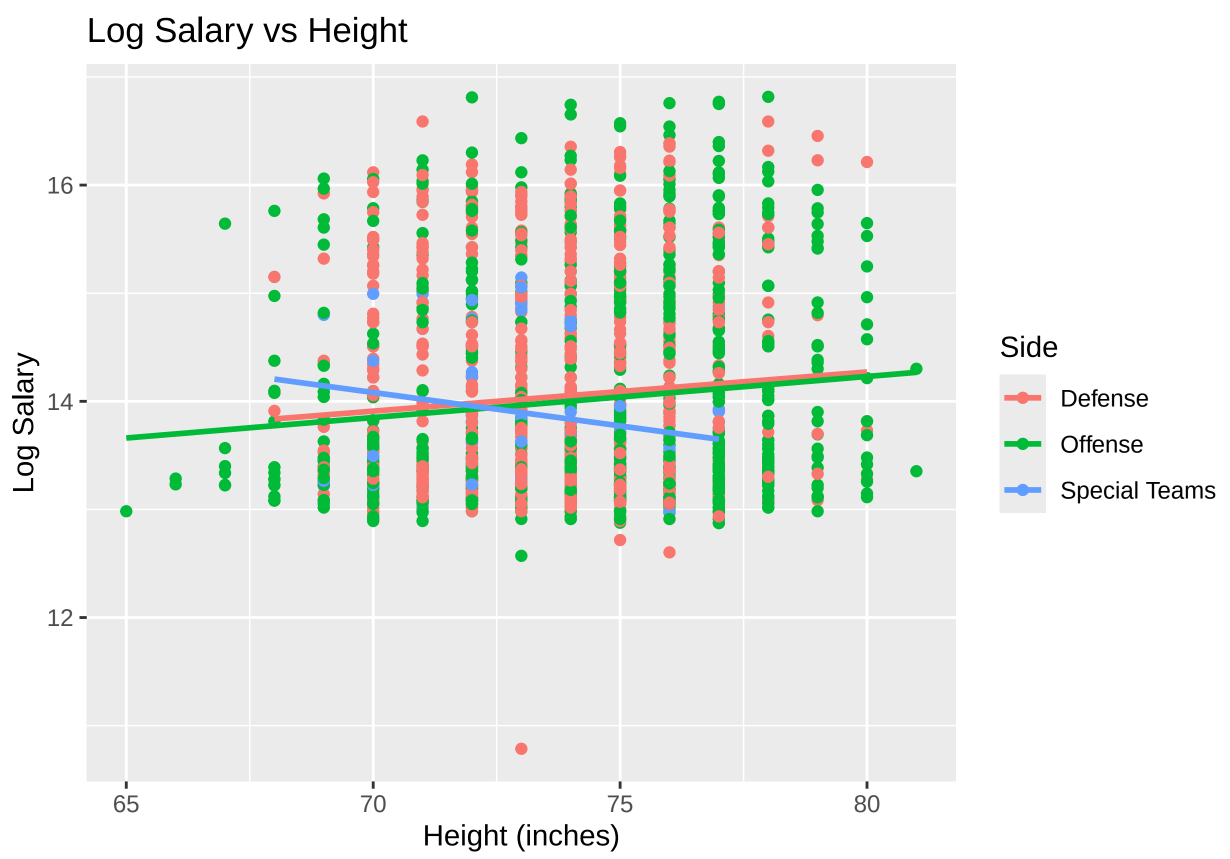
## ~~~ Visualizations ~~~

### 7. Make a graph with 3 variables: X axis, Y axis, and plot point coloring, that tells a story.

**please note:** we will be deducting points for messy plots, plots with poor labeling, or plots that are hard to read. Make sure your plots are clear and easy to understand.

```
# You pick the variables.  What would be interesting?

# Combine player positions to create three "Side" categories: "Offense", "Defense", and "Special Teams"

clean.dat <- clean.dat %>%
    mutate( Side = case_when(
        Position %in% c("C", "FB", "OL", "QB", "RB", "TE", "WR") ~ "Offense",
        Position %in% c("CB", "DE", "DL", "LB", "NT", "S") ~ "Defense",
        Position %in% c("K", "LS", "P") ~ "Special Teams",
        TRUE ~ NA_character_
    ) )

# Plot with the new `Side` variable instead of `Position`
ggplot( clean.dat, aes( x = total_inches, y = logSal, color = Side ) ) +
    geom_point() +
    geom_smooth( method = "lm", se = FALSE ) +
    labs( title = "Log Salary vs Height",
          x = "Height (inches)",
          y = "Log Salary" ) +
    theme(legend.position = "right")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



**7(B)** Explain what the plot tells us.

Our plot shows how different sides of the game might have different relationships between height and salary. The trend lines indicate that taller players who are on offense and defense make more money, while special teams players' salaries appear to be higher for shorter players.

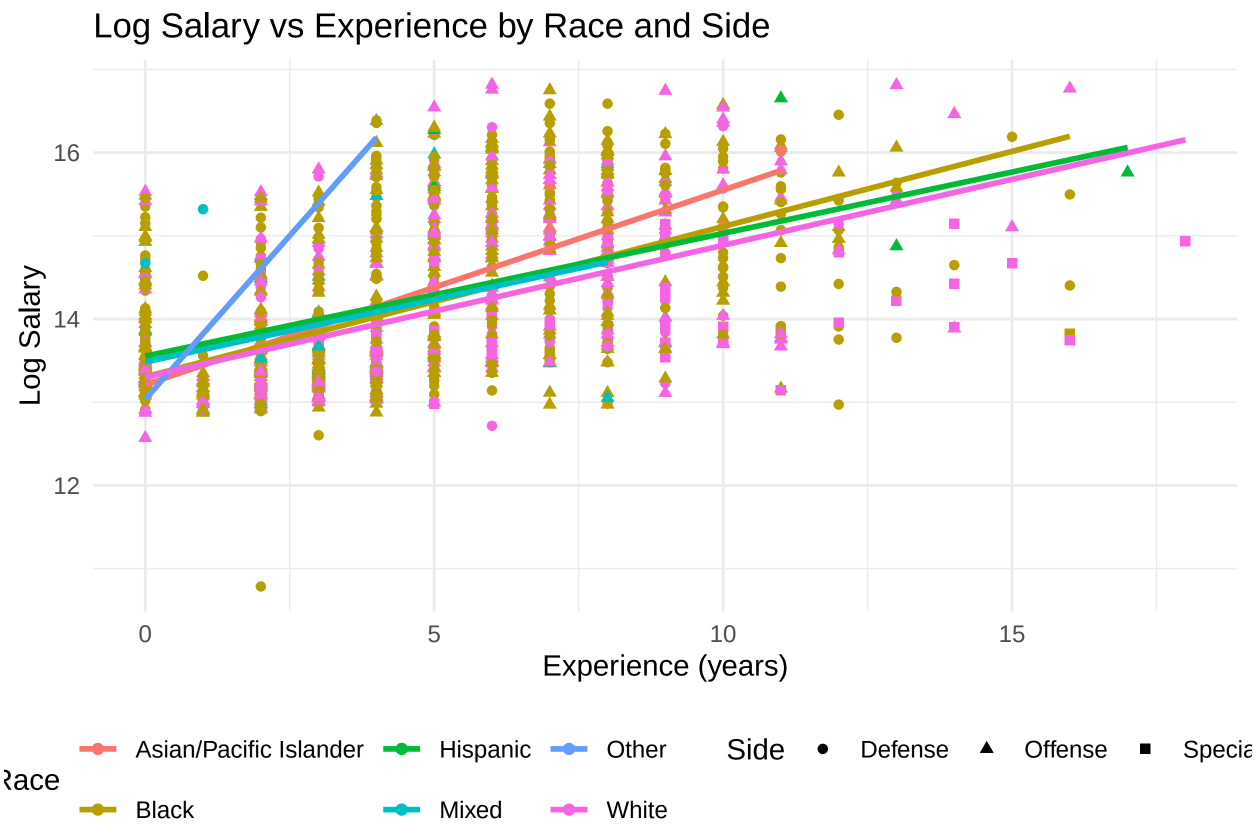
## 8. Make a graph with 4 variables that tells a story that your first plot does not.

For the four variables, perhaps you could use X axis, Y axis, plot point coloring, and plot point size, or perhaps plot point shape or some other attribute. **Do not** use any facets yet. Just a single plot. Other than that constraint, you pick!

```
# We will make a plot that explores how log salary, experience, race, and side (offense, defense, speci

# Remove NA from Race column

clean.dat <- clean.dat %>%
    filter( !is.na( Race ) )

# Combine Asian/Pacific Islander and Pacific Islander into one category called Asian/Pacific Islander,

clean.dat <- clean.dat %>%
  mutate(Race = if_else(Race == "Pacific Islander",
                        "Asian/Pacific Islander",
                        Race)) %>%
    mutate(Race = if_else(Race %in% c("Native American", "Other", "Iranian"), "Other", Race))
```

```r
ggplot(clean.dat, aes(x = Experience, y = logSal, color = Race)) +
  geom_point(aes(shape = Side)) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Log Salary vs Experience by Race and Side",
    x = "Experience (years)",
    y = "Log Salary",
    shape = "Side"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

```
## 'geom_smooth()' using formula = 'y ~ x'

## Warning: Removed 2 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 2 rows containing missing values or values outside the scale range
## ('geom_point()').
```



**8(B)** What does this plot tell us that the other plot doesn't?

This plot shows that (as expected), salary increases with years of experience. Hispanic, White, and Black show a similar trend, while Asian/Pacific Islander seems to show a steeper relationship (where salary increases more for each year of experience). Other races show a very steep relationship, however, this is likely due to a small sample size rather than a statistically significant relationship. It is difficult to tell from the graph if Side relates to the other variables, though it appears that potentially, offensive players make more money than others. However, the graph is very busy and overly complicated.

### 9. When might we prefer simpler plots with fewer moving pieces to more complicated plots?

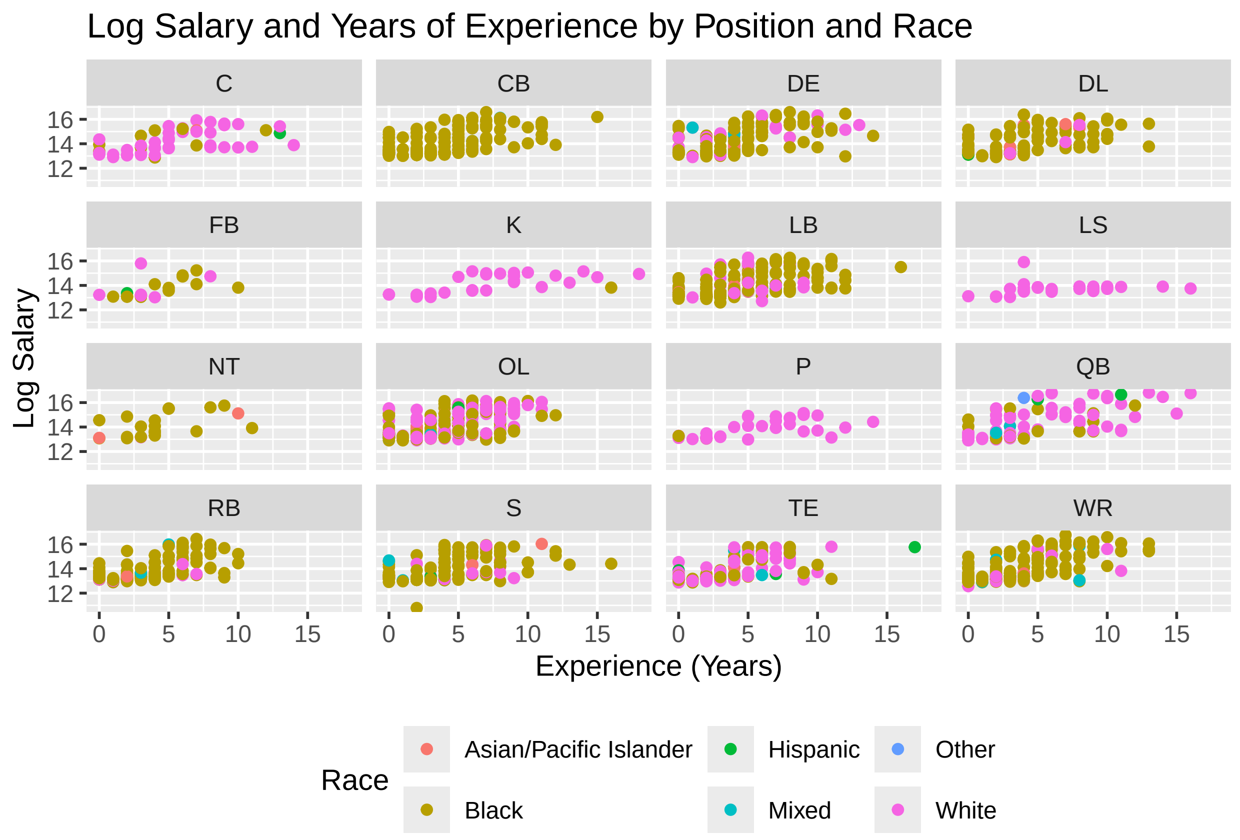
The purpose of a plot is to convey information: often, certain types of information are better articulated visually than through words. When making a plot, the user should always be kept in mind — what should they take away from the plot? What is the intended message? If a plot is too complicated and includes too many variables, this message can become obscured and difficult to understand. Often, the most effective plots are simple and allow the user to intuitively understand what they are supposed to be taking away from the plot. The plot above is overly complicated and makes it difficult to draw clear conclusions from the data.

### 10. Now extend or modify your plot by making multiple plots in the same window.

You want to use either `facet_wrap()` or `facet_grid()` here.

```r
# We modified our plot to see how years of experience, race, salary, and position are related. Using fa

ggplot( clean.dat, aes( x = Experience, y = logSal, color = Race ) ) +
    facet_wrap(~Position) +
    geom_point() +
    labs( title = "Log Salary and Years of Experience by Position and Race",
          x = "Experience (Years)",
          y = "Log Salary") +
    theme(legend.position = "bottom")
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```



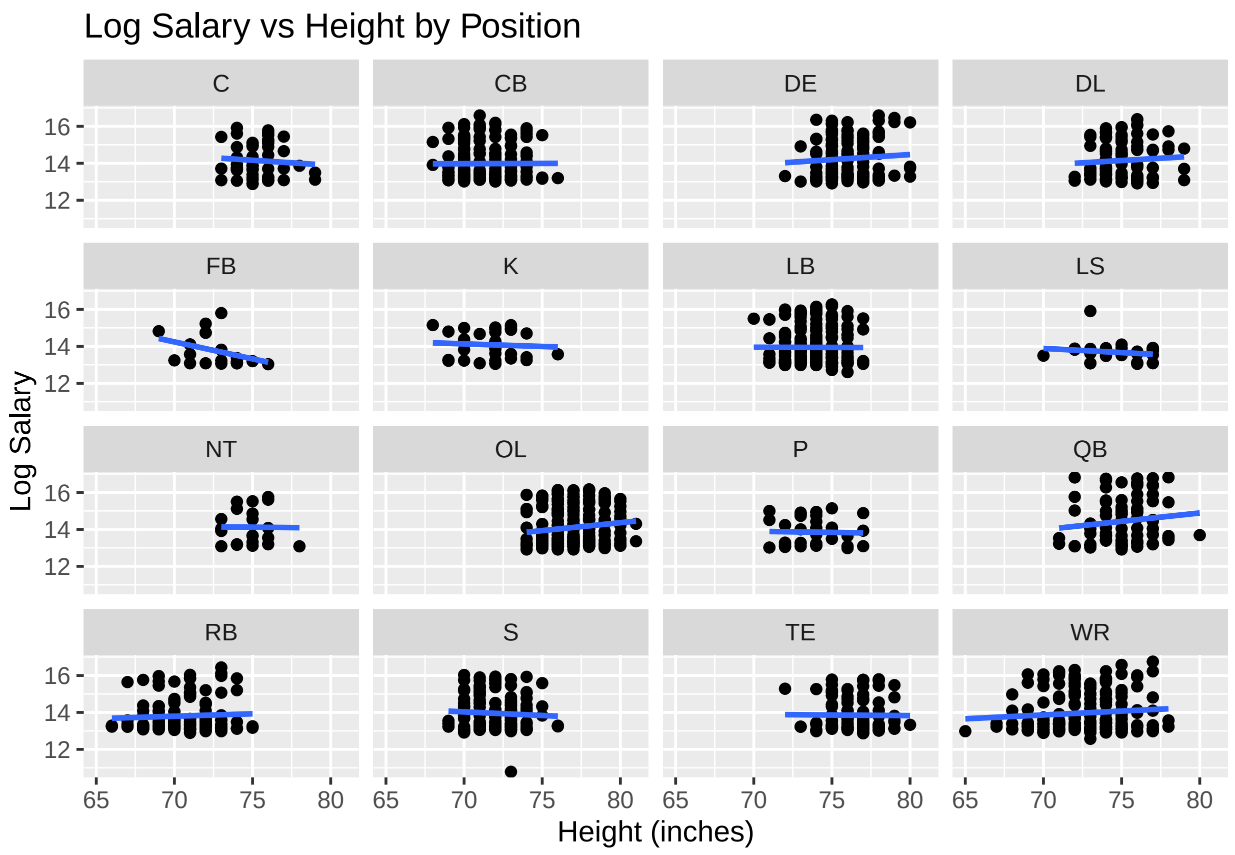
**10(B)** Compared to your prior plots, does this plot make anything easier to see? Harder to see?

This plot makes the data, and the conclusions we can draw from it, much clearer. The color coding by race shows some very close relationships between race and position, and for many positions, there is a clear correlation between years of experience and log salary. We can also see a relationship between position and years of experience — positions where players are injured more often (such as running back) show shorter career lengths than other positions.

## 11. I'd like to know if taller players are paid more in some positions but not others. Make a plot to answer that question.

```r
ggplot( clean.dat, aes( x = total_inches, y = logSal)) +
    facet_wrap(~Position) +
    geom_point() +
    geom_smooth( method = "lm", se = FALSE ) +
    labs( title = "Log Salary vs Height by Position",
          x = "Height (inches)",
          y = "Log Salary" ) +
    theme(legend.position = "bottom")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



**11(B).** What does your plot say?

Our plot shows that there appears to be a relationship between height and log salary for some positions, such as fullback, offensive lineman, and quarterback, but not for others. The plot also shows that certain positions have a wider distribution of heights than others — wide receivers are of many heights, whereas all offensive linemen in the data set are tall. We can also see that certain positions, such as quarterback, make more money on average than others.

## ~~~ A touch of modeling ~~~

**12. (2 pts) Make a predictive model (using OLS) to predict log salary using some set of other covariates and print out a summary of the model. Specify how you decided to treat missing data.**

```
# Create a model that predicts log salary based on experience, age, and position.

# I was cleaning na when fitting the model, so I got an error saying: ! Can't recycle input of size 174

clean2 <- na.omit(clean.dat)

model_final <- lm(
  logSal ~ Experience + Age + Position,
  data = clean2,
  na.action = na.omit)

clean2$pred_logSal <- predict(model_final)

summary( model_final )
```

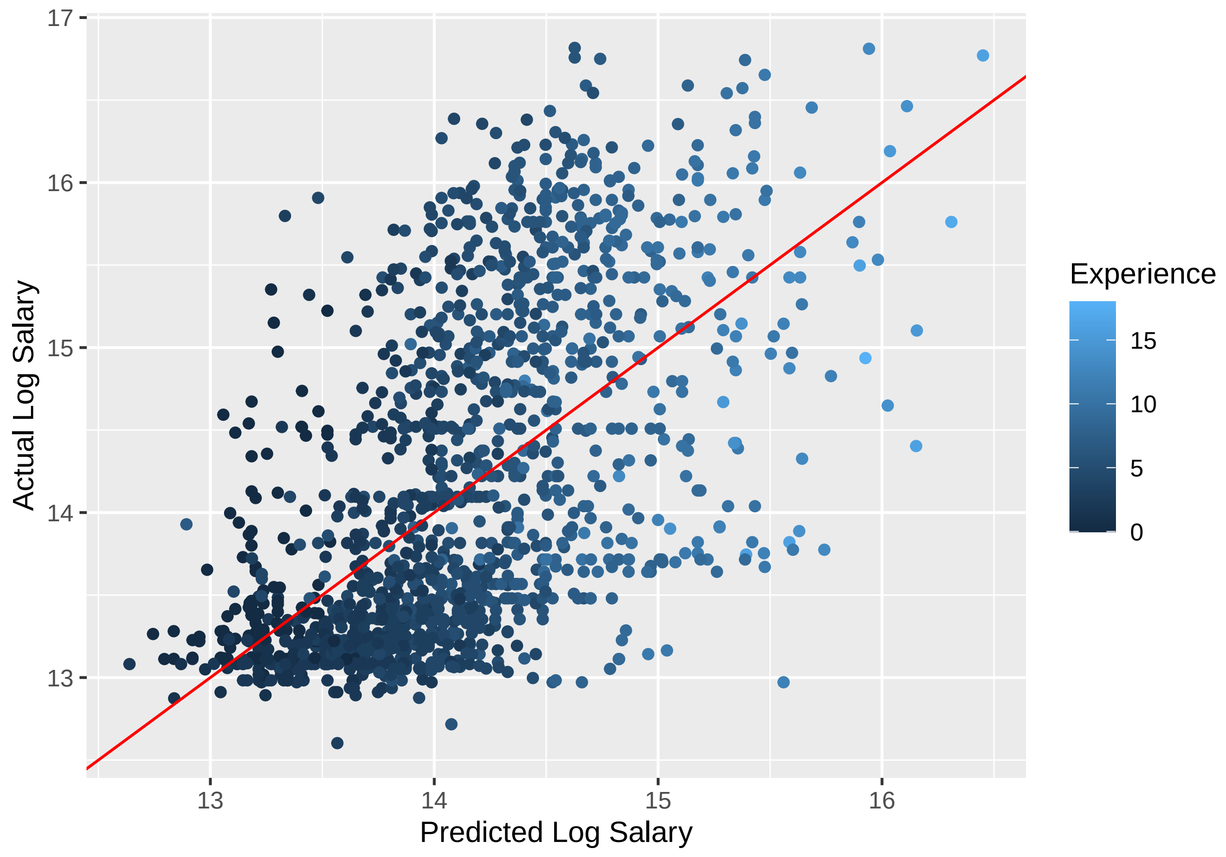
```
##
## Call:
## lm(formula = logSal ~ Experience + Age + Position, data = clean2,
##     na.action = na.omit)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5888 -0.5680 -0.1813  0.5362  2.4646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.27173    0.45594  35.688  < 2e-16 ***
## Experience   0.29556    0.01756  16.832  < 2e-16 ***
## Age         -0.12577    0.01788  -7.036 2.99e-12 ***
## PositionCB  -0.01611    0.13365  -0.121  0.90408
## PositionDE   0.26958    0.13634   1.977  0.04820 *
## PositionDL   0.15595    0.14474   1.077  0.28146
## PositionFB  -0.17786    0.20934  -0.850  0.39565
## PositionK   -0.38358    0.18849  -2.035  0.04202 *
## PositionLB  -0.06965    0.12972  -0.537  0.59140
## PositionLS  -0.57682    0.19018  -3.033  0.00246 **
## PositionNT   0.02756    0.22115   0.125  0.90083
## PositionOL   0.08671    0.12893   0.673  0.50136
## PositionP   -0.29097    0.19016  -1.530  0.12619
## PositionQB   0.35539    0.14787   2.403  0.01636 *
## PositionRB  -0.05097    0.13861  -0.368  0.71313
## PositionS   -0.06903    0.13686  -0.504  0.61405
## PositionTE  -0.08150    0.14121  -0.577  0.56392
## PositionWR   0.04792    0.13264   0.361  0.71795
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.7963 on 1524 degrees of freedom
## Multiple R-squared:  0.3585, Adjusted R-squared:  0.3513
## F-statistic:  50.1 on 17 and 1524 DF,  p-value: < 2.2e-16
```

Missing data: At first, when trying to fit the model, we used na.action = na.omit within the lm() function. But R couldn't run the predict() function, because there was incompatibility with the number of rows ( 1743 x 1745). So to ensure we were predicting the log salary of the same individuals, we decided to remove null values of the entire dataset within this specific chunk of code, using: `clean2 <- na.omit(clean.dat)` before fitting the model.

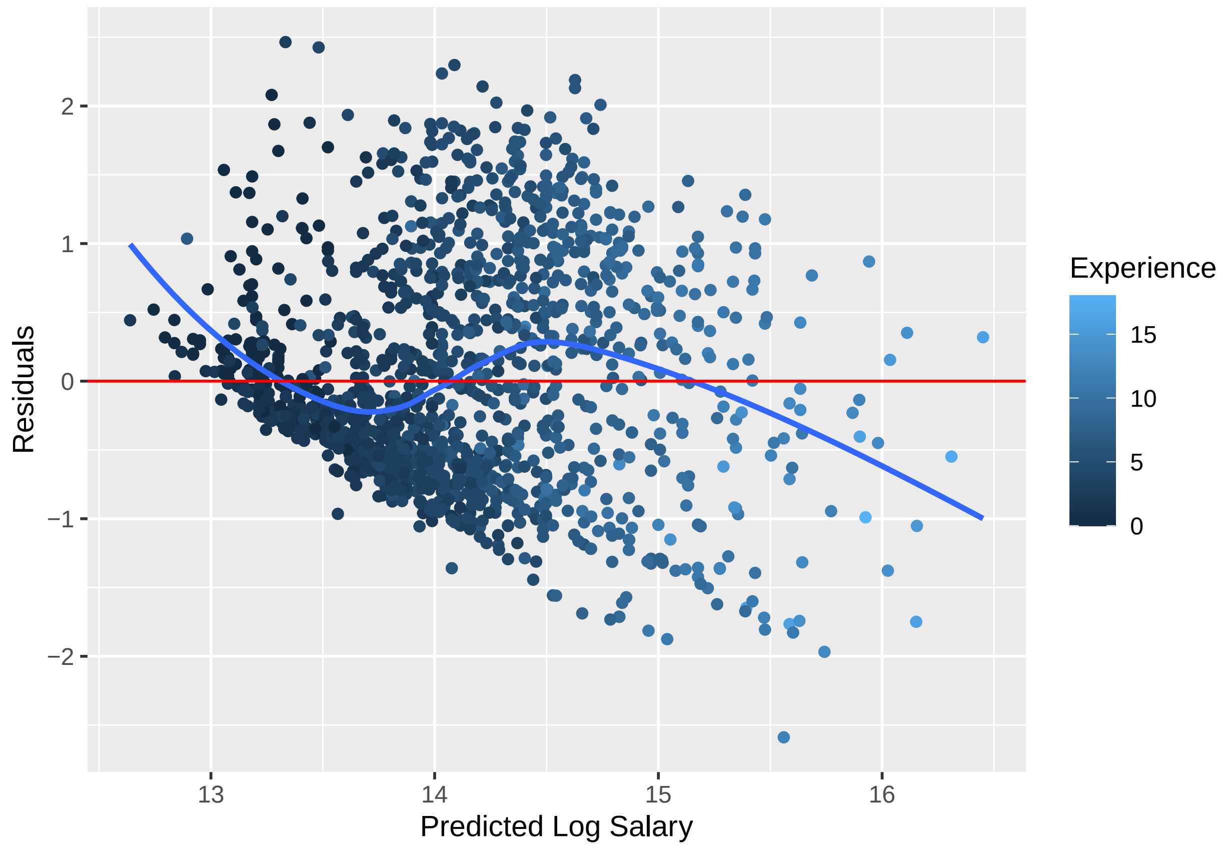
## 13. Make a plot showing predictions vs actual, and also make a residual plot.

```r
clean2 <- clean2 %>%
  filter(complete.cases(Experience, Age, Position))

# the above line was suggested by chatgpt after failing to filter na in multiple variables.

# model predicted vs actual
ggplot( clean2, aes( x = predict( model_final ), y = logSal , color = Experience) ) +
    geom_point() +
    geom_abline( slope = 1, intercept = 0, col = "red" ) +
    labs( y = "Actual Log Salary", x = "Predicted Log Salary" )
```



```r
# residuals plot
ggplot( clean2, aes( x = predict( model_final ), y = resid( model_final ), color = Experience ) ) +
    geom_point() +
    geom_smooth( method = "loess", se = FALSE ) +
    geom_hline( yintercept = 0, col = "red" ) +
    labs( x = "Predicted Log Salary", y = "Residuals" )
```

```
## 'geom_smooth()' using formula = 'y ~ x'

## Warning: The following aesthetics were dropped during statistical transformation:
## colour.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?
```



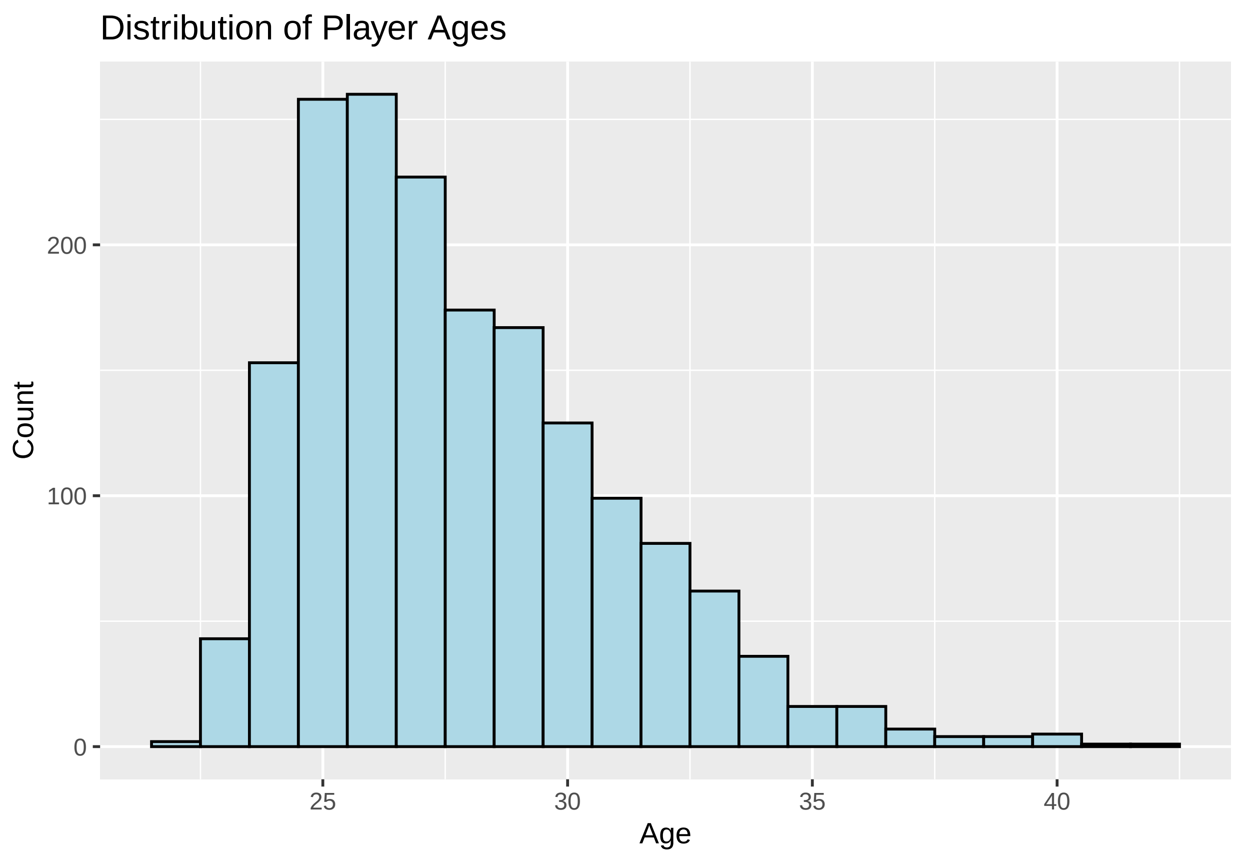
**13(B)** Do these plots show any problems?

The plot showing the relationship between predicted and actual salary does not show a good fit according to the linear model. The residuals plot shows a slanted pattern, which also indicates that our model does not fit the data well (the residuals should be randomly clustered around zero).

## 14. How old do players get?

Make a distribution of the ages of players in the dataset, and then speculate about how long a player might be able to play in the NFL.

```r
# Histogram of player ages

clean.dat %>%
    ggplot( aes( x = Age ) ) +
    geom_histogram( binwidth = 1, fill = "lightblue", color = "black" ) +
    labs( title = "Distribution of Player Ages",
          x = "Age",
          y = "Count" )
```



```r
# Let's use the 75th percentile to estimate the age of retirement for NFL players.

quantile( clean.dat$Age, probs = 0.75, na.rm = TRUE )
```

```
## 75%
##  30
```

```r
summary( clean.dat$Age, na.rm = TRUE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   22.00   25.00   27.00   27.82   30.00   42.00
```

If we take the 75% percentile as a possible indication of the age of retirement, this is estimated to be 30 years old.

**14(B).** Poke around on the internet to see what a typical age of retirement is. How did that compare to your own assessment?

The average age of retirement in the NFL is roughly 27.6 years. (Source: https://runrepeat.com/nfl-player-career-length)

Interestingly, this aligns almost exactly with the mean and median age in our dataset. Would this have perhaps been a better way of estimating retirement age than using the 75% percentile?

## 15. (2 pts) Use these data and your work to predict what average lifetime earnings would be for a median level pay NFL player.

```
# We calculated median lifetime salary by position, and then took a weighted average. Reasoning is expl

# According to statista.com (https://www.statista.com/statistics/240102/average-player-career-length-in

# We need to reconcile these positions with the positions in our dataset, with help from ChatGPT to cla

table(clean.dat$Position)
```

```
##
##   C  CB  DE  DL  FB   K  LB  LS  NT  OL   P  QB  RB   S  TE  WR
##  51 176 142 102  21  30 232  29  21 243  33  84 135 142 112 192
```

```
# Not clearly indicated on the list from statista.com:
# C   DE   FB   LS  NT     S
# Center, Defensive End, Fullback, Long Snapper, Nose Tackle, Safety

# List from statista.com, updated with missing positions based on ChatGPT's advice:
# Kickers/punters (K, P, LS): 4.87 years
# Quarterbacks (QB): 4.44
# Offensive Linemen (OL, C): 3.63
# Defensive Linemen (DL, DE, NT): 3.24
# Linebackers (LB): 2.97
# Cornerbacks (CB, S): 2.94
# Tight Ends (TE): 2.85
# Wide receivers (WR): 2.81
# Running backs (RB, FB): 2.57
# Average across all positions: 3.3

# Create a new variable in my dataset that assigns the average career length to each player based on th

clean.dat <- clean.dat %>%
    mutate( Career_Length = case_when(
        Position %in% c("K", "P", "LS") ~ 4.87,
        Position == "QB" ~ 4.44,
        Position %in% c("OL", "C") ~ 3.63,
        Position %in% c("DL", "DE", "NT") ~ 3.24,
        Position == "LB" ~ 2.97,
        Position %in% c("CB", "S") ~ 2.94,
        Position == "TE" ~ 2.85,
        Position == "WR" ~ 2.81,
        Position %in% c("RB", "FB") ~ 2.57,
        TRUE ~ NA_real_
    ) )

# Now we need the median salary for each position.

clean.dat %>%
    group_by( Position ) %>%
    summarise( median_salary = median( Salary, na.rm = TRUE ) )
```

```
## # A tibble: 16 x 2
##    Position median_salary
##    <chr>            <dbl>
##  1 C              1050000
##  2 CB              724666.
##  3 DE             1177492
##  4 DL              983836
##  5 FB              636300
##  6 K              1275000
##  7 LB              780000
##  8 LS              905000
##  9 NT             1105000
## 10 OL              840000
## 11 P               905000
## 12 QB             1267986
## 13 RB              698140
## 14 S               800000
## 15 TE              654488.
## 16 WR              685032.
```

```
# Put this data into a new data frame

position_earnings <- clean.dat %>%
    group_by( Position ) %>%
    summarise( median_salary = median( Salary, na.rm = TRUE ),
               career_length = first( Career_Length ),
               count = n() ) %>%
    mutate( lifetime_earnings = median_salary * career_length )


# Now we need a weighted average of lifetime earnings by position, based on the proportion of positions

position_earnings %>%
    summarize(avg_lifetime = sum( (lifetime_earnings * count )/ sum( count )) )
```

```
## # A tibble: 1 x 1
##   avg_lifetime
##          <dbl>
## 1     2814507.
```

```
# Average lifetime earnings for a median level pay NFL player would be approximately $2,814,507.
```

## ~~~ Extensions ~~~

In this part of the assignment we offer two, very open ended, extensions. They represent a reasonable amount of work, but are only worth 2 points—in other words, you can choose to do them if you want to go above and beyond, but you can still get a fine grade without doing them at all. We will only grade at most one extension, and we will grade the first we see.

Both extensions relate to a second dataset, `Master-Player-List-2024.csv`, that you have in the data folder.

### Extension Option 1. Comparing to a more modern dataset.

How much more are players paid now, compared to 2013? Try to account for differences in the datasets in terms of, for example, age and experience of the players in each dataset. Write up your results using at least one visualization and some text explaining any analysis you did.

```
nfl_24 <- read_csv( "data/Master-Player-List-2024.csv",
                 na = c("NA","N/A","", "n/a", "#VALUE!") )
```

```
## Rows: 2846 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): PlayerID, Name, Team, Position, Height, DOB, YrsExp, CapHit
## dbl (2): Weight, Age
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Removing NAs from CapHit in 2024 dataset
clean.dat.24 <- nfl_24 %>%
    filter( !is.na( CapHit ) )

# Creating Salary variable in 2024 dataset from CapHit
clean.dat.24 <- clean.dat.24 %>%
    mutate( Salary = parse_number( CapHit ) )

# Finding the middle and average salary in both 2013 and 2024
clean.dat %>%
    filter( !is.na( Salary ) ) %>%
    summarise( mean_2013 = mean( Salary ),
               median_2013 = median( Salary ) )
```

```
## # A tibble: 1 x 2
##   mean_2013 median_2013
##       <dbl>       <dbl>
## 1  2142266.      815375
```

```
clean.dat.24 %>%
    filter( !is.na( Salary ) ) %>%
    summarise( mean_2024 = mean( Salary ),
               median_2024 = median( Salary ) )
```

```
## # A tibble: 1 x 2
##   mean_2024 median_2024
##       <dbl>       <dbl>
## 1  2720630.    1045108.
```

```
# Calculating the mean difference in pay between salaries in 2024 and 2013
pay_diff = mean( clean.dat.24$Salary ) - mean( clean.dat$Salary )
pay_diff
```

```
## [1] 578364.6
```

On average, players in 2024 are making $578,365 per year more than players in 2013.

```
# Data cleaning
# Converting YrsExp variable in 2024 dataset to numeric, treating "R" as 0 years of experience
clean.dat.24 <- clean.dat.24 %>%
  mutate(
    Experience = ifelse(YrsExp == "R", "0", YrsExp),
    Experience= as.numeric(Experience) )

# Cleaning position in 2024 dataset and creating Side variable similar to 2013 dataset
clean.dat.24 <- clean.dat.24 %>%
    mutate( Side = case_when(
        Position %in% c("C", "FB", "OT", "OG", "QB", "RB", "C", "TE", "WR", "OL") ~ "Offense",
        Position %in% c("DB", "LB", "DE", "DT", "DL","CB", "NT", "SS", "FS", "OLB", "MLB", "ILB", "SAF")
        Position %in% c("K", "LS", "P") ~ "Special Teams",
        TRUE ~ NA_character_ ) )

# Cleaning position in 2024 dataset so that the positions are similar to 2013 dataset
clean.dat.position <- clean.dat.24 %>%
  mutate( Position = case_when(
      Position %in% c("OT", "OG", "OL") ~ "OL",
      Position == "C"                   ~ "C",
      Position == "FB"                  ~ "FB",
      Position == "QB"                  ~ "QB",
      Position == "RB"                  ~ "RB",
      Position == "TE"                  ~ "TE",
      Position == "WR"                  ~ "WR",
      Position %in% c("DB", "CB")       ~ "CB",
      Position %in% c("SS", "FS", "SAF") ~ "S",
      Position %in% c("LB", "OLB", "MLB", "ILB") ~ "LB",
      Position %in% c("DT", "DL")       ~ "DL",
      Position == "NT"                  ~ "NT",
      Position == "DE"                  ~ "DE",
      Position == "K"                  ~ "K",
      Position == "P"                  ~ "P",
      Position == "LS"                  ~ "LS",
      TRUE                              ~ NA_character_ ) )
```

```
# Creating one dataset that has the mean salary for each experience level in both 2013 and 2024, so we
exp.sum <- bind_rows(
  clean.dat %>%
    group_by(Experience) %>%
    summarise(mean_salary = mean(Salary, na.rm = TRUE) ) %>%
    mutate(year = "2013"),
  clean.dat.24 %>%
    group_by(Experience) %>%
    summarise(mean_salary = mean(Salary, na.rm = TRUE) ) %>%
    mutate(year = "2024") )

# Calculating the mean salary difference and percent change between 2013 and 2024 for player experience
exp.sum %>%
  group_by( Experience ) %>%
  summarise( mean_salary_2013 = mean( mean_salary[ year == "2013" ] ),
             mean_salary_2024 = mean( mean_salary[ year == "2024" ] ) ) %>%
  mutate( salary_diff = mean_salary_2024 - mean_salary_2013 ) %>%
  mutate ( percent_diff = salary_diff / mean_salary_2013 * 100 ) %>%
  arrange( desc( percent_diff ) )
```

***Visualization 1: Plotting Average Salary differences between 2013 and 2024 based on years of experience***

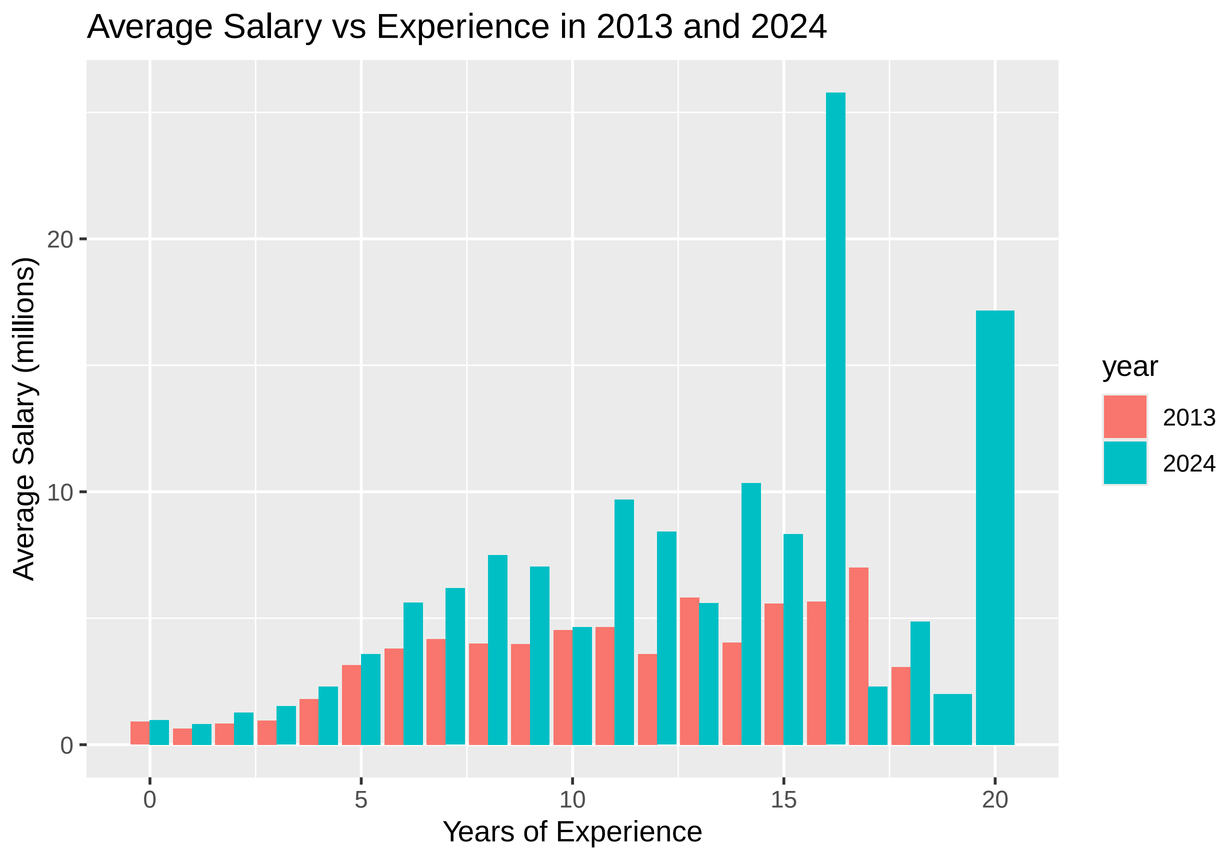
```
## # A tibble: 22 x 5
##    Experience mean_salary_2013 mean_salary_2024 salary_diff percent_diff
##         <dbl>            <dbl>            <dbl>       <dbl>        <dbl>
##  1         16          5662500         25775000    20112500        355.
##  2         14         4033167.        10346583.    6313417.        157.
##  3         12         3585953.         8418285.    4832332.        135.
##  4         11         4659902.         9687686.    5027784.        108.
##  5          8         4008626.         7504232.    3495607.         87.2
##  6          9         3989331.         7051626.    3062295.         76.8
##  7         18          3066667          4875000     1808333         59.0
##  8          3          957115.         1520888.     563773.         58.9
##  9          2          835412.         1271973.     436561.         52.3
## 10         15          5575000         8322202.    2747202.         49.3
## # i 12 more rows
```

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ggplot(exp.sum, aes(x = Experience, y = mean_salary, fill = year)) +
  geom_col(position = "dodge") +
  scale_y_continuous( labels = function(x) x / 1e6,
                      name   = "Average Salary (millions)" ) +
  labs( title = "Average Salary vs Experience in 2013 and 2024",
    x = "Years of Experience" )
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_col()‘).
```



The first visualization, Average Salary vs Experience in 2013 and 2024, displays a bar chart with the average salary (in millions of USD) on the y-axis and how many years a player has been playing in the league on the x-axis. Overall, average salaries are higher in 2024 at almost every experience level with players in 2024 making, on average, roughly $578,365 more per year than players in 2013. We see two noticeable spikes in this visualization at years 16 and 20 which are due to two quarterbacks, Matthew Stafford and Aaron Rodgers, respectively. In 2013, there were no players in the NFL playing in years 19 and 20.

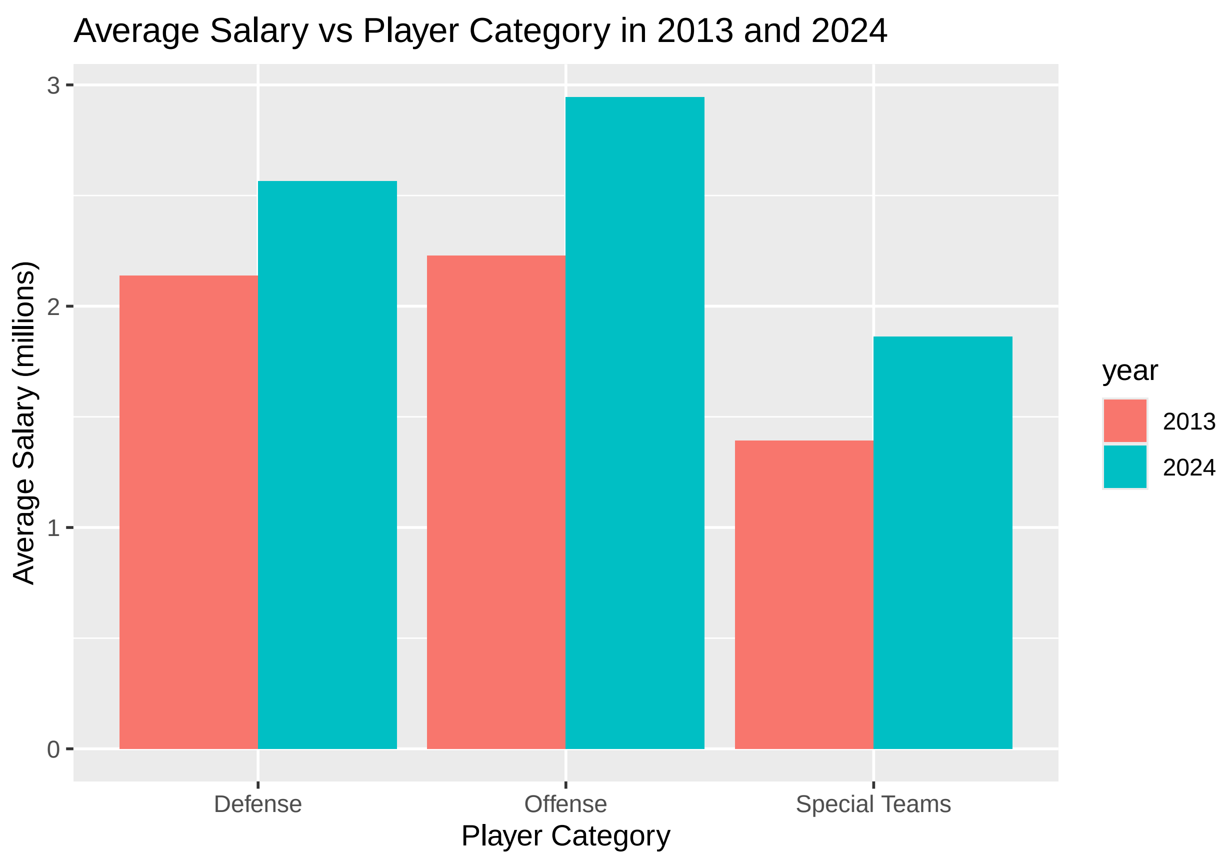
To create this first visualization, we first parsed the 2024 CapHit variable to make it comparable to the 2013 Salary variable. In the 2024 dataset, we turned the years of experience into a numeric variable after converting R to 0 to represent 0 years of experience. We then combined the 2013 and 2024 dataset into a single dataset, grouped by years of experience, and calculated the average salary for each group. These averages were then plotted against years of experience for both years.

```
# Creating one dataset that has the mean salary for each side of the ball (offense, defense, special te
side.sum <- bind_rows(
  clean.dat %>%
    group_by(Side) %>%
    summarise(mean_salary = mean(Salary, na.rm = TRUE) ) %>%
    mutate(year = "2013"),
  clean.dat.24 %>%
    group_by(Side) %>%
    summarise(mean_salary = mean(Salary, na.rm = TRUE) ) %>%
    mutate(year = "2024") )

# Calculating the mean salary difference and percent change between 2013 and 2024 for each side of the
side.sum %>%
  group_by( Side ) %>%
  summarise( mean_salary_2013 = mean( mean_salary[ year == "2013" ] ),
             mean_salary_2024 = mean( mean_salary[ year == "2024" ] ) ) %>%
  mutate( salary_diff = mean_salary_2024 - mean_salary_2013 ) %>%
  mutate ( percent_diff = salary_diff / mean_salary_2013 * 100 ) %>%
  arrange( desc( percent_diff ) )
```

***Visualization 2: Plotting Average Salary differences between 2013 and 2024 based on side of the ball***

```
## # A tibble: 3 x 5
##   Side          mean_salary_2013 mean_salary_2024 salary_diff percent_diff
##   <chr>                    <dbl>            <dbl>       <dbl>        <dbl>
## 1 Special Teams         1391979.         1863275.     471296.         33.9
## 2 Offense               2228981.         2946056.     717075.         32.2
## 3 Defense               2137798.         2565532.     427734.         20.0
```

```
# Plotting Average Salary differences between 2013 and 2024 based on side of the ball
ggplot(side.sum, aes(x = Side, y = mean_salary, fill = year)) +
  geom_col(position = "dodge") +
  scale_y_continuous( labels = function(x) x / 1e6,
                      name   = "Average Salary (millions)" ) +
  labs( title = "Average Salary vs Player Category in 2013 and 2024",
    x = "Player Category" )
```



The second visualization, Average Salary vs Side of the Ball in 2013 and 2024, displays a bar chart with the average salary (in millions of USD) on the y-axis and the player category (offense, defense, or special teams) on the x-axis. We see that the average salary for each player category increased from 2013 to 2024, with the largest percent change coming to special teams players at ~34% salary increase. Offense and defense also increased by ~32% and ~20%, respectively.

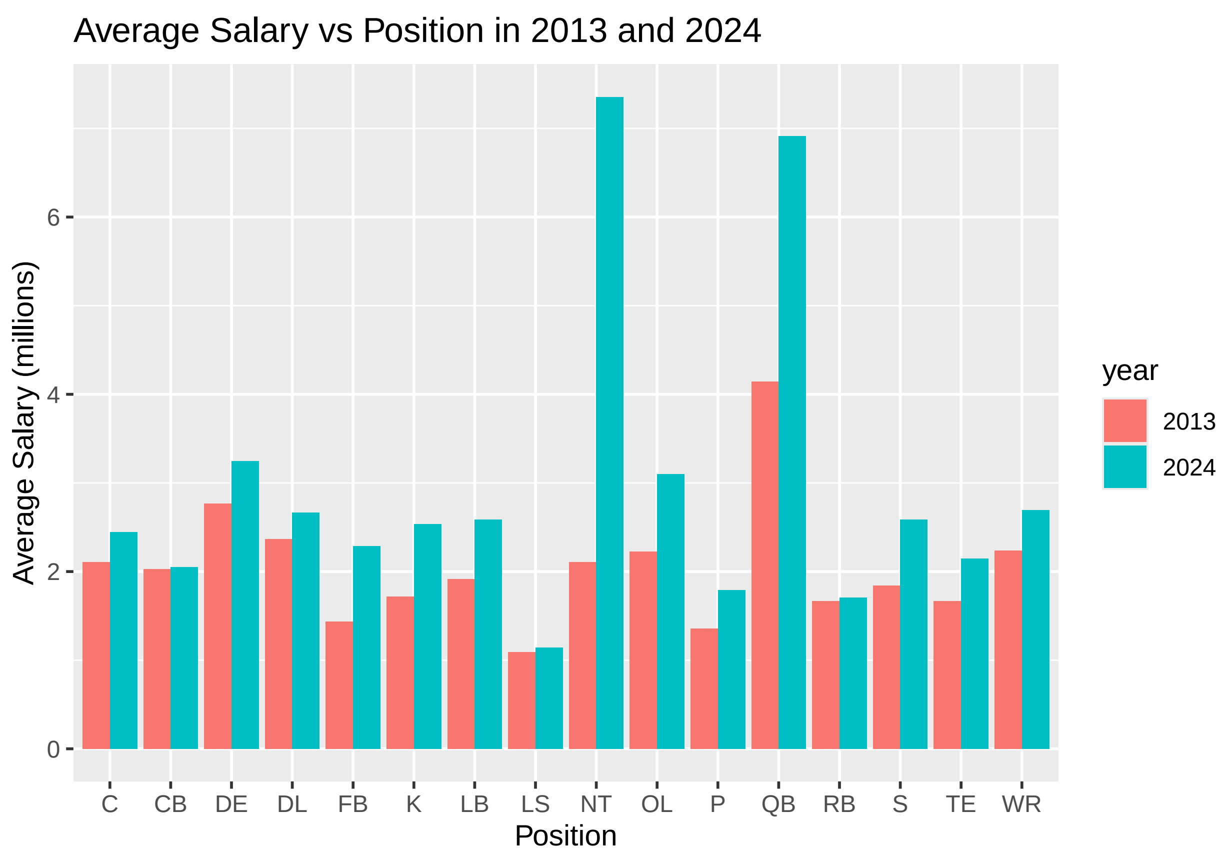
To create this second visualization, we classified all of the player positions into three categories for both the 2013 and 2024 datasets. We then combined the datasets and created a summary dataset, side.sum, and calculated the average salary of each new category. These averages were then plotted as side-by-side bars for 2013 and 2024.

```
# Creating one dataset that has the mean salary for each position in both 2013 and 2024, so we can comp
position.sum <- bind_rows(
  clean.dat %>%
    group_by(Position) %>%
    summarise(mean_salary = mean(Salary, na.rm = TRUE) ) %>%
    mutate(year = "2013"),
  clean.dat.position %>%
    group_by(Position) %>%
    summarise(mean_salary = mean(Salary, na.rm = TRUE) ) %>%
    mutate(year = "2024") )

# Calculating the mean salary difference and percent change between 2013 and 2024 for each position
position.sum %>%
  group_by( Position ) %>%
  summarise( mean_salary_2013 = mean( mean_salary[ year == "2013" ] ),
             mean_salary_2024 = mean( mean_salary[ year == "2024" ] ) ) %>%
  mutate( salary_diff = mean_salary_2024 - mean_salary_2013 ) %>%
  mutate ( percent_diff = salary_diff / mean_salary_2013 * 100 ) %>%
  arrange( desc( percent_diff ) )
```

***Visualization 3: Plotting Average Salary differences between 2013 and 2024 based on position***

```
## # A tibble: 16 x 5
##    Position mean_salary_2013 mean_salary_2024 salary_diff percent_diff
##    <chr>               <dbl>            <dbl>       <dbl>        <dbl>
##  1 NT               2107992.         7356394.    5248402.      249.
##  2 QB               4145334.         6911429.    2766096.       66.7
##  3 FB               1437786.         2290181      852395.       59.3
##  4 K                1719945          2534668.     814723.       47.4
##  5 S                1843162.         2588004.     744842.       40.4
##  6 OL               2223450.         3102289.     878839.       39.5
##  7 LB               1917398.         2586395.     668997.       34.9
##  8 P                1357487.         1792170.     434683.       32.0
##  9 TE               1666640.         2144300.     477660.       28.7
## 10 WR               2238717.         2694527.     455810.       20.4
## 11 DE               2767078.         3244449.     477371.       17.3
## 12 C                2107085.         2447746.     340661.       16.2
## 13 DL               2366991.         2663080.     296090.       12.5
## 14 LS               1091954.         1143052.      51098.        4.68
## 15 RB               1668350.         1705040.      36690.        2.20
## 16 CB               2029060.         2049226.      20167.        0.994
```

```
# Plotting Average Salary differences between 2013 and 2024 based on position
ggplot(position.sum, aes(x = Position, y = mean_salary, fill = year)) +
  geom_col(position = "dodge") +
  scale_y_continuous( labels = function(x) x / 1e6,
                      name   = "Average Salary (millions)" ) +
  labs( title = "Average Salary vs Position in 2013 and 2024",
    x = "Position" )
```



The third visualization, Average Salary vs Position in 2013 and 2024, displays a bar chart with the average salary (in millions of USD) on the y-axis and player position on the x-axis. We see that the average salary for each position increased from 2013 to 2024. Most notably, the positions of nosetackle (NT) and quarterback (QB) both see large increases. Nosetackles increased their average salary from 2013 to 2024 by ~250% followed by quarterbacks at ~67%. While there are many quarterbacks in the NFL (84 in 2013 and 118 in 2024) there are significantly fewer nosetackles (21 in 2013 and 9 in 2024). Because there are so few nosetackles, two players' contracts significantly increase the average salary for the position.

To create this third visualization, we had to ensure that all player positions were identical in both 2013 and 2024. This was done by using the position types in the 2013 dataset as a reference and changing the position types in the 2024 dataset to match via the mutate function. One more dataset was created, position.sum, which grouped players by position and year and calculated the average salary for each position in 2013 and 2024. These averages were plotted as side-by-side bars for each year at each position.