

# final\_proj\_qmd

## AUTHOR

Imogen Meers

## How do combine measurements affect draft prospects?

---

### Why?

I chose this problem to look for correlation between combine measurements and draft results because the draft is so competitive and athletes/coaches are always looking for ways to get an edge. If there were particular measurements you could target to improve your chances that would be useful for athletes/coaches to know.

### How?

#### 1. Using a binomial logistic regression to predict combine result of drafted or not

I created a binomial field to indicate whether all combine athletes were drafted or not, which I would use as an outcome for my binomial logistic regression. I started testing with 1 predictor using each of the most popular combine measurements. Then, I compared the top 2 using main effect terms and interaction and the top 3 using main effect terms. I made sure to use VIF values and check for drastic changes in coefficients of the same variables to ensure there was not too much collinearity between my predictors. I made a table of sorted AIC values and ANOVA to compare which model was the best.

#### 2. Effect of weight by position on pick number

After testing for general correlation, I created a mixed effects model to see effect of weight on pick number when creating an intercept by position. I chose to include the position as in football, body types differ a lot and depending on the position, it is common for players to have different weight goals.

NB: I chose to only look at weight (not hand, arm and height) because whilst interesting, are not that useful for prospects as they cannot be changed.

### Data

Loading relevant packages and data

```
library(dplyr)
library(readr)
library(lme4)
library(ggplot2)
library(tidyverse)
library(corr)
```

```
library(sjPlot)

combine <- read_csv("C:/Users/immmim/Downloads/combine.csv")
draft <- read_csv("C:/Users/immmim/Downloads/draft.csv")
```

Clean and create usable dataset "combine\_draft"

```
draft <- draft %>%
  mutate(draft = as.numeric(draft)) %>%
  select(playerId, draft, round, pick) %>%
  filter(draft >= 1987)

combine_draft <- combine %>% distinct(playerId, .keep_all = TRUE) %>%
  full_join(draft, by = "playerId")

combine_draft <- combine_draft %>% mutate(drafted = ifelse(!is.na(draft), 1, 0))

abbreviation = c("QB", "RB", "FB", "TB", "HB", "OL", "G", "LG", "RG", "T", "LT", "RT", "C")
position = c("Quarterback", "Running Back", "Fullback", "Tailback", "Halfback", "Offensive Tackle",
           "Outside Linebacker", "Cornerback", "Safety", "Strong Safety", "Free Safety")

#so that i can use full name for position when i graph the intersect
lookup <- as.data.frame(list(abbreviation, position))
colnames(lookup) = c("combinePosition", "fullPos")

combine_draft <- inner_join(combine_draft, lookup)
```

## Visualisations to select predictors

1. Using binomial model to predict effect on drafted or not

2. Effect of weight by pos on round number

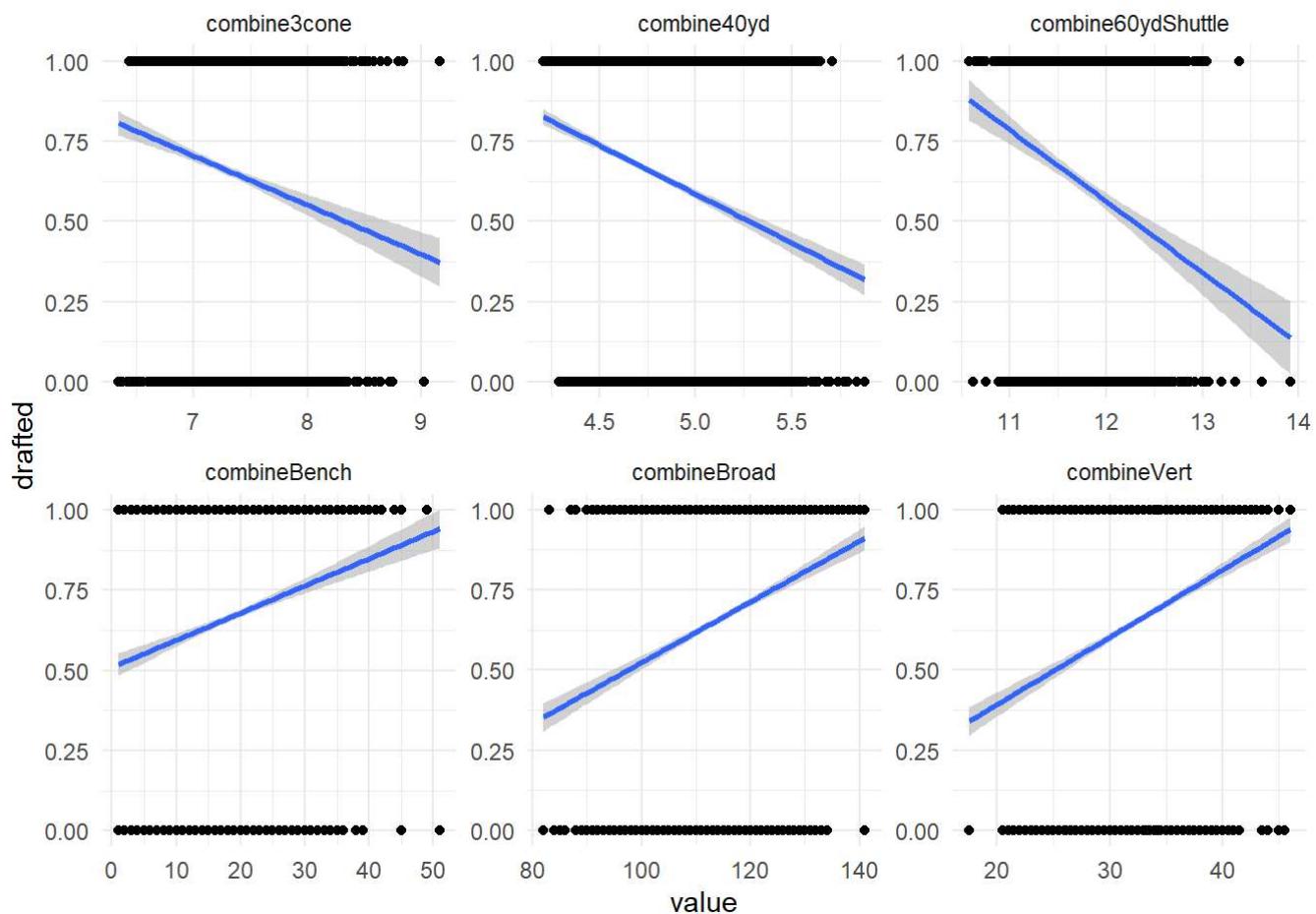
```
combine_draft %>%
  # Select just the variables that
  # I want to visualize:
  select(drafted, combine40yd,
         combineVert, combineBroad, combineBench, combine3cone, combine60ydShuttle) %>%
  # Use the whiff column as an "ID" column
  # and then pivot the data into long form
  # I'd use whiffs as my outcome variable,
  # so I want to see everything compared to
  # it. If you are planning to use a different
  # variable as your outcome, you should change
  # what comes after the - to that variable.
  pivot_longer(cols = -drafted) %>%
  # Plot every value against whiffs
```

```
ggplot(aes(value, drafted)) +
  geom_point() +
  geom_smooth(method = "lm") +
  # Put each variable into its own plot:
  facet_wrap(vars(name), scales = "free") +
  theme_minimal()
```

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

Warning: Removed 18121 rows containing non-finite outside the scale range  
(`stat\_smooth()`).

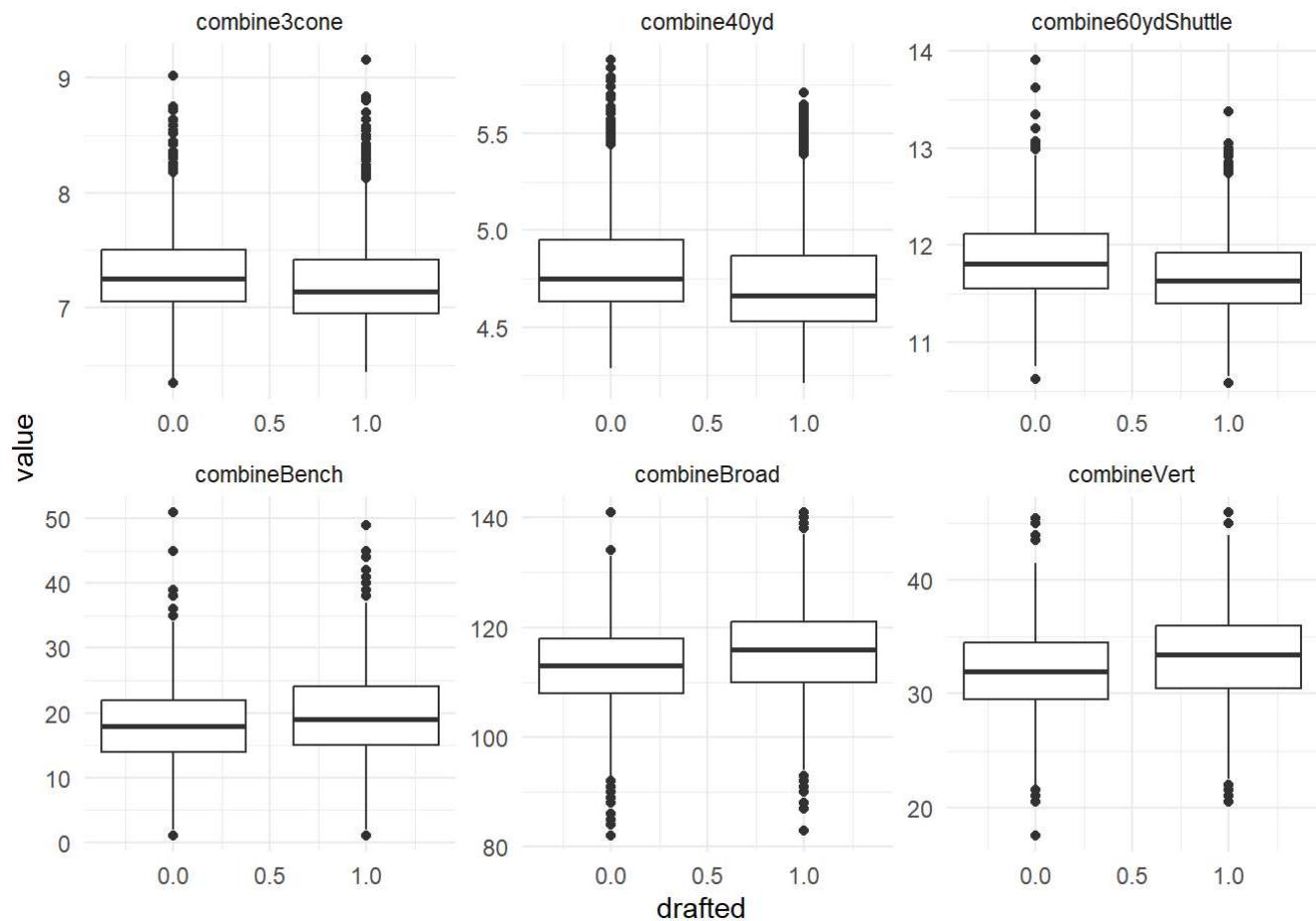
Warning: Removed 18121 rows containing missing values or values outside the scale range  
(`geom\_point()`).



```
combine_draft %>%
  # Select just the variables that
  # I want to visualize:
  select(drafted, combine40yd,
         combineVert, combineBroad, combineBench, combine3cone, combine60ydShuttle) %>%
  # Use the whiff column as an "ID" column
  # and then pivot the data into long form
  # I'd use whiffs as my outcome variable,
```

```
# so I want to see everything compared to
# it. If you are planning to use a different
# variable as your outcome, you should change
# what comes after the - to that variable.
pivot_longer(cols = -drafted) %>%
# Plot every value against whiffs
ggplot() +
geom_boxplot(aes(x=drafted, y = value, group=drafted)) +
# Put each variable into its own plot:
facet_wrap(vars(name), scales = "free") +
theme_minimal()
```

Warning: Removed 18121 rows containing non-finite outside the scale range  
(`stat\_boxplot()`).



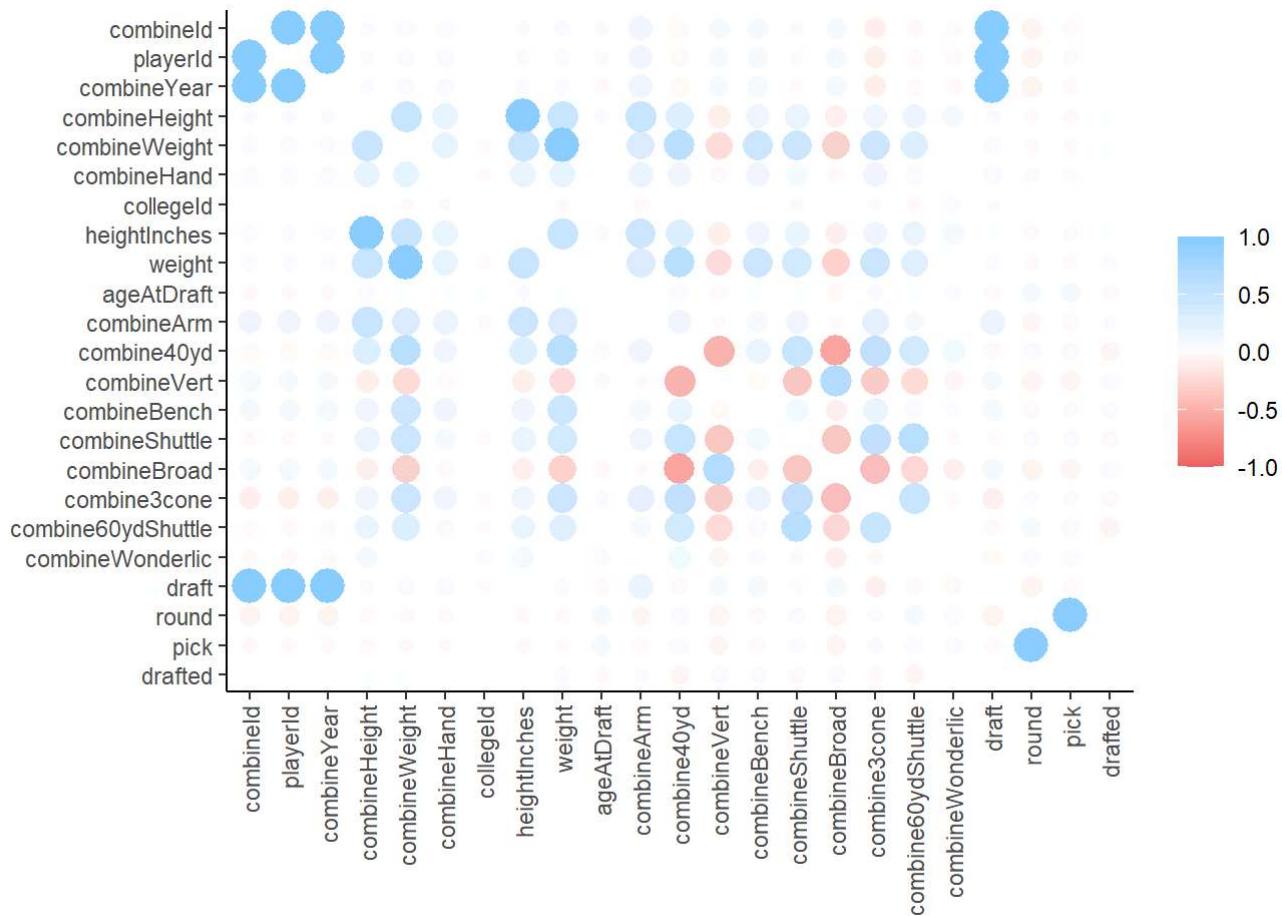
```
rplot(correlate(combine_draft))+ theme(axis.text.x = element_text(angle = 90, vjust = 0))
```

Non-numeric variables removed from input: `combinePosition`, `nameFirst`, `nameLast`, `nameFull`, `position`, `nflId`, `college`, `dob`, `playerProfileUrl`, `homeCity`, `homeState`, `homeCountry`, `highSchool`, `hsCity`, `hsState`, `hsCountry`, and `fullPos`

Warning in stats::cor(x = x, y = y, use = use, method = method): the standard deviation is zero

Correlation computed with

- Method: 'pearson'
- Missing treated using: 'pairwise.complete.obs'



## Models for part 1

### 1 Main Effect Term

```
model1 <- glm(drafted ~ combineBroad , data = combine_draft, family = binomial)
summary(model1)
```

Call:

```
glm(formula = drafted ~ combineBroad, family = binomial, data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-4.199748	0.359155	-11.69	<2e-16 ***
combineBroad	0.042669	0.003157	13.51	<2e-16 ***

```
-->
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

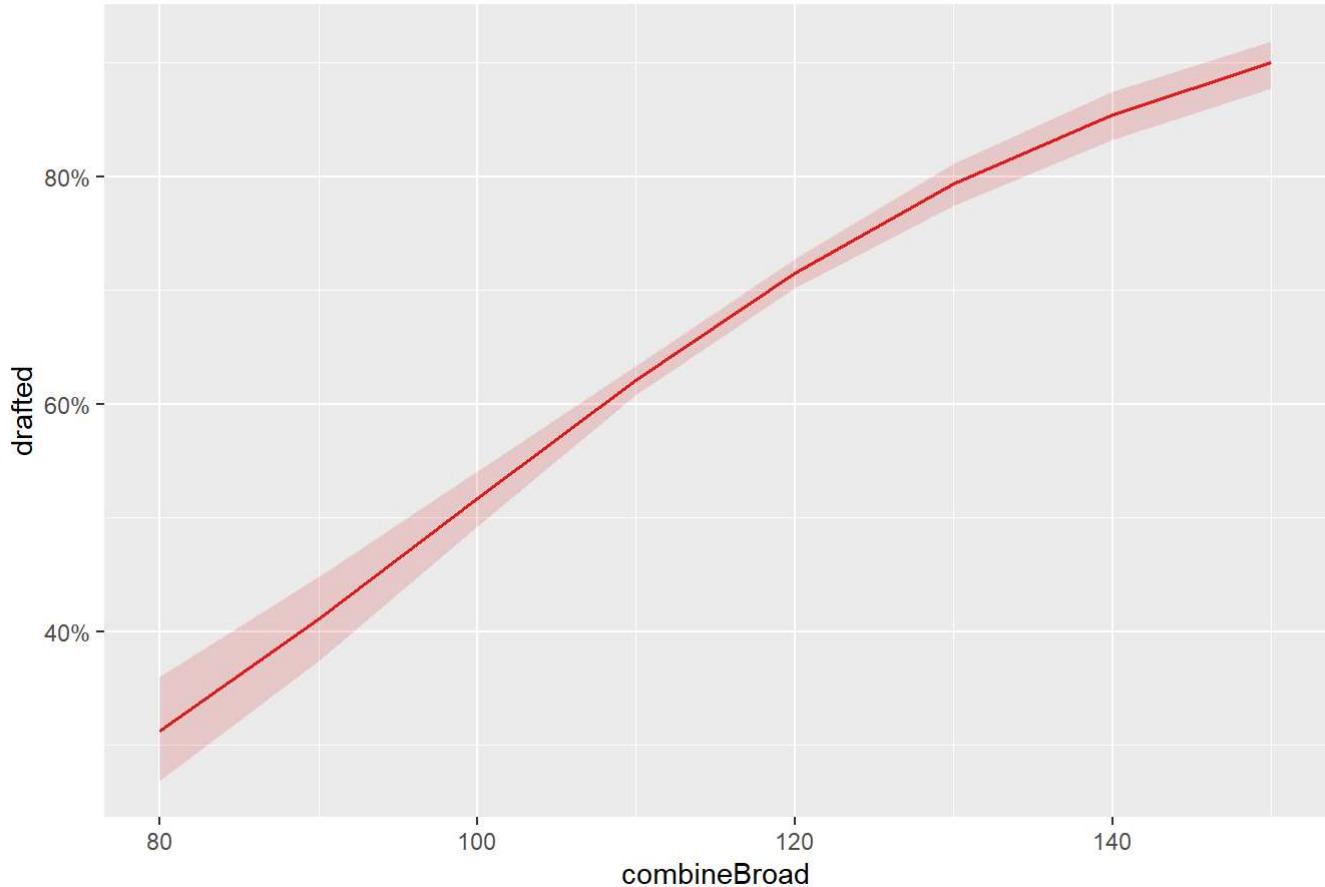
Null deviance: 9173.6 on 7146 degrees of freedom  
 Residual deviance: 8984.9 on 7145 degrees of freedom  
 (1528 observations deleted due to missingness)  
 AIC: 8988.9

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model1$terms)[3], AIC(model1))
aic_models <- as.data.frame(x, col.names = c("model", "aic"))
sjPlot::plot_model(model1, type = "pred") #part of logistic regression curve
```

Data were 'prettified'. Consider using `terms="combineBroad [all]"` to  
 get smooth plots.

### Predicted probabilities of drafted



```
model1_2 <- glm(drafted ~ combineVert , data = combine_draft, family = binomial)
summary(model1_2)
```

Call:

```
glm(formula = drafted ~ combineVert, family = binomial, data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.459462	0.219389	-11.21	<2e-16 ***
combineVert	0.096026	0.006732	14.26	<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: 9287.7 on 7253 degrees of freedom

Residual deviance: 9075.8 on 7252 degrees of freedom

(1421 observations deleted due to missingness)

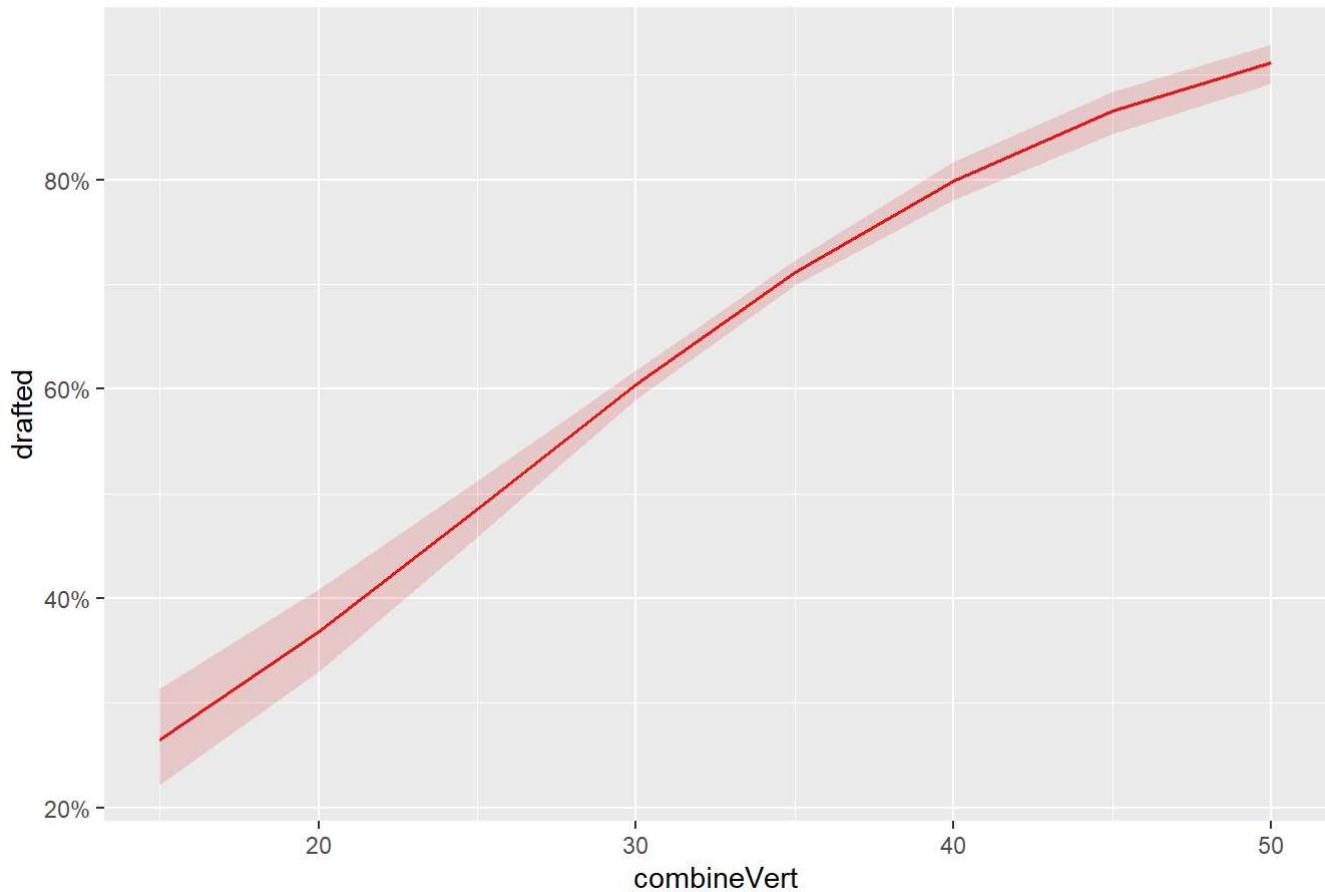
AIC: 9079.8

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model1_2$terms)[3], AIC(model1_2))
aic_models <- rbind(aic_models, x)
sjPlot::plot_model(model1_2, type ="pred") #part of logistic regression curve
```

Data were 'prettified'. Consider using `terms="combineVert [all]"` to  
get smooth plots.

## Predicted probabilities of drafted



```
model1_3 <- glm(drafted ~ combine40yd , data = combine_draft, family = binomial)
summary(model1_3)
```

Call:

```
glm(formula = drafted ~ combine40yd, family = binomial, data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	7.05124	0.45949	15.35	<2e-16 ***
combine40yd	-1.33949	0.09613	-13.93	<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: 9875.4 on 7716 degrees of freedom

Residual deviance: 9678.0 on 7715 degrees of freedom

(958 observations deleted due to missingness)

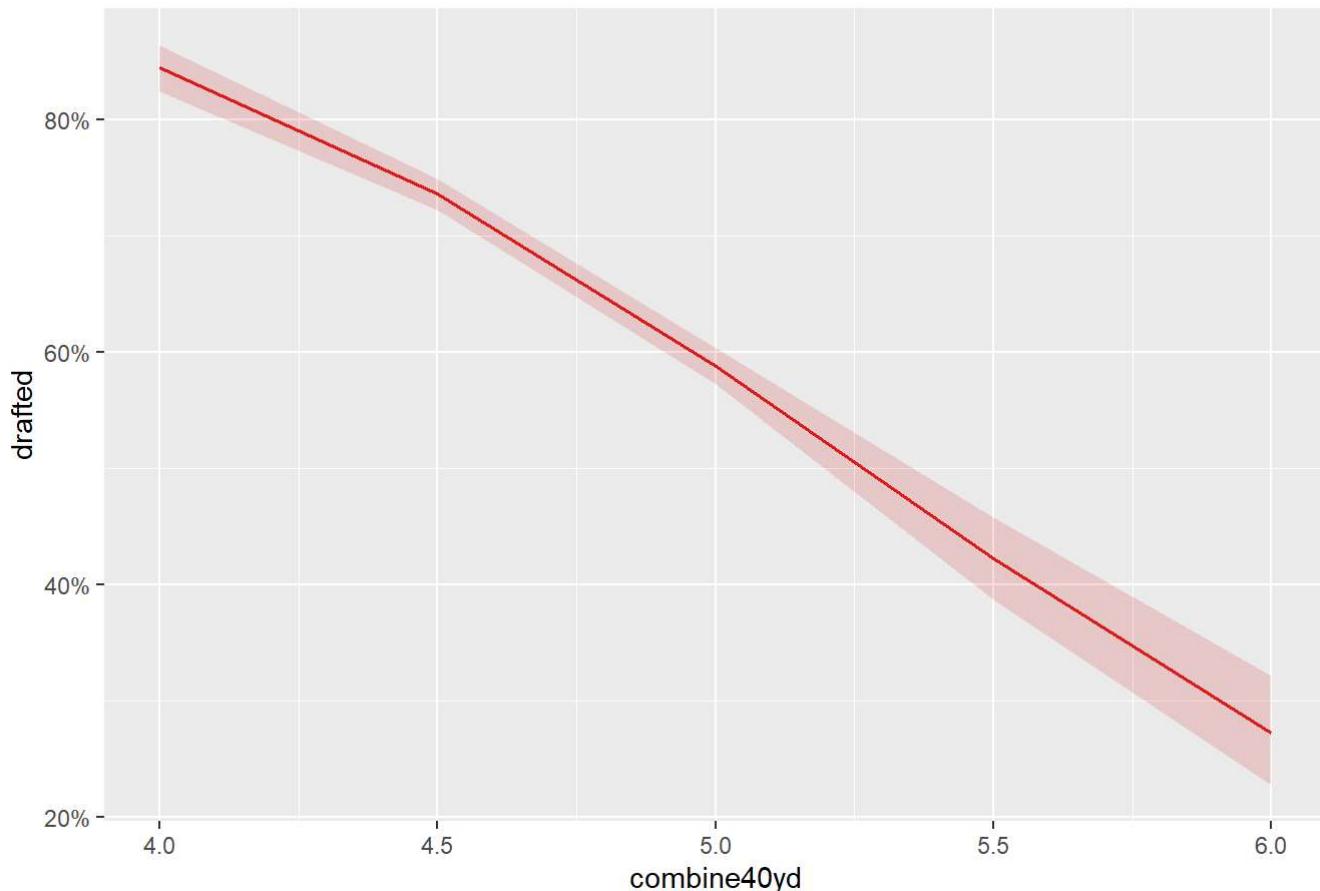
AIC: 9682

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model1_3$terms)[3], AIC(model1_3))
aic_models <- rbind(aic_models, x)
sjPlot::plot_model(model1_3, type = "pred") #part of logistic regression curve
```

Data were 'prettified'. Consider using `terms="combine40yd [all]"` to get smooth plots.

### Predicted probabilities of drafted



```
model1_4 <- glm(drafted ~ combine60ydShuttle , data = combine_draft, family = binomial)
summary(model1_4)
```

Call:

```
glm(formula = drafted ~ combine60ydShuttle, family = binomial,
  data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	11.8793	1.4066	8.446	< 2e-16 ***
combine60ydShuttle	-0.9681	0.1194	-8.108	5.15e-16 ***
---				
Signif. codes:	0 ****	0.001 **	0.01 *	0.05 .
	'	'	'	'

(Dispersion parameter for binomial family taken to be 1)

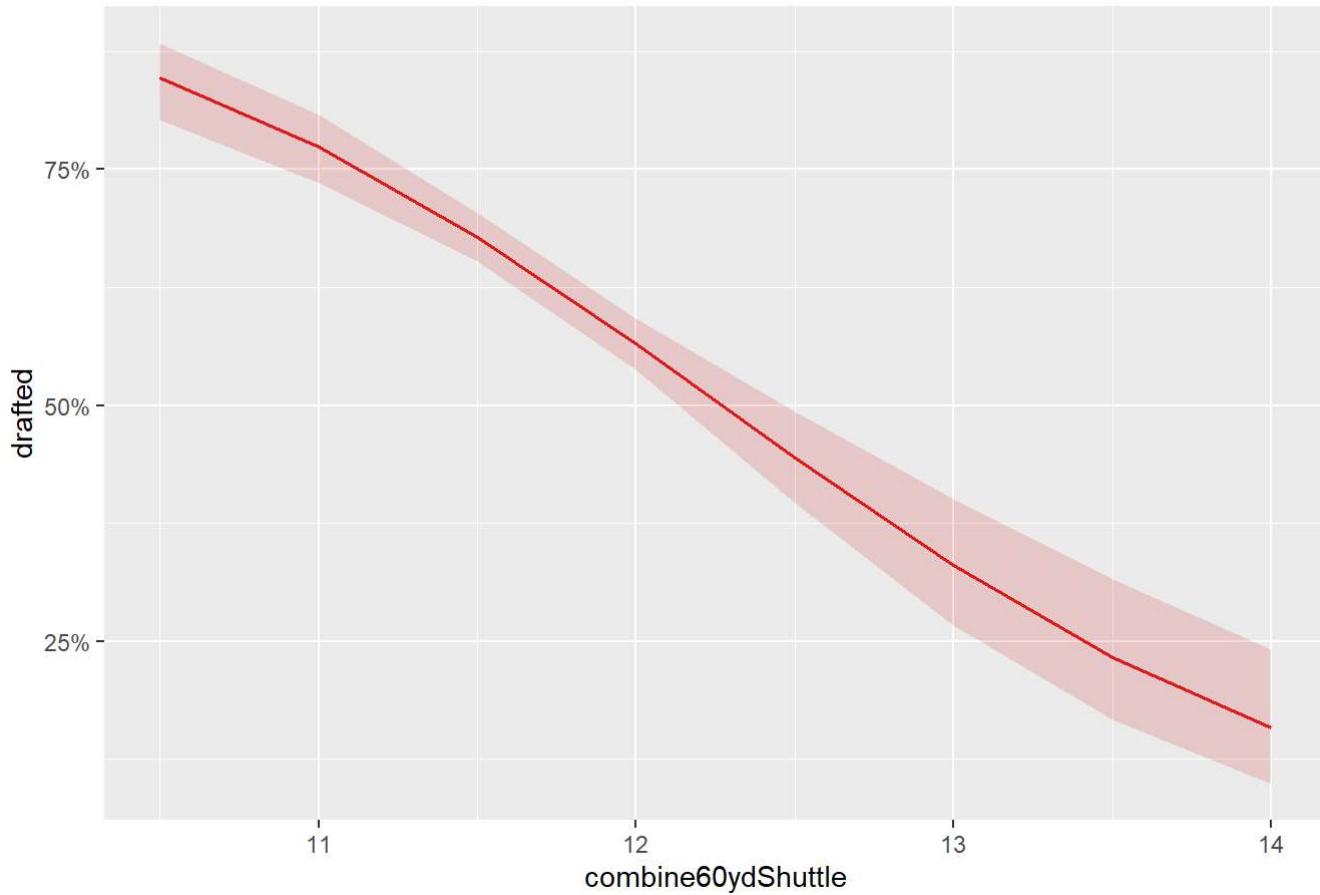
Null deviance: 2345.3 on 1767 degrees of freedom  
 Residual deviance: 2275.9 on 1766 degrees of freedom  
 (6907 observations deleted due to missingness)  
 AIC: 2279.9

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model1_4$terms)[3], AIC(model1_4))
aic_models <- rbind(aic_models, x)
sjPlot::plot_model(model1_4, type = "pred") #part of logistic regression curve
```

Data were 'prettified'. Consider using `terms="combine60ydShuttle [all]"` to get smooth plots.

### Predicted probabilities of drafted



```
model1_5 <- glm(drafted ~ combineBench, data = combine_draft, family = binomial)
summary(model1_5)
```

Call:

glm(formula = drafted ~ combineBench, family = binomial, data = combine\_draft)

## Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-0.026631	0.086185	-0.309	0.757
combineBench	0.039383	0.004403	8.945	<2e-16 ***
---				
Signif. codes:	0 ****	0.001 **	0.01 *	0.05 .
	'	'	'	'
	'	'	'	'

(Dispersion parameter for binomial family taken to be 1)

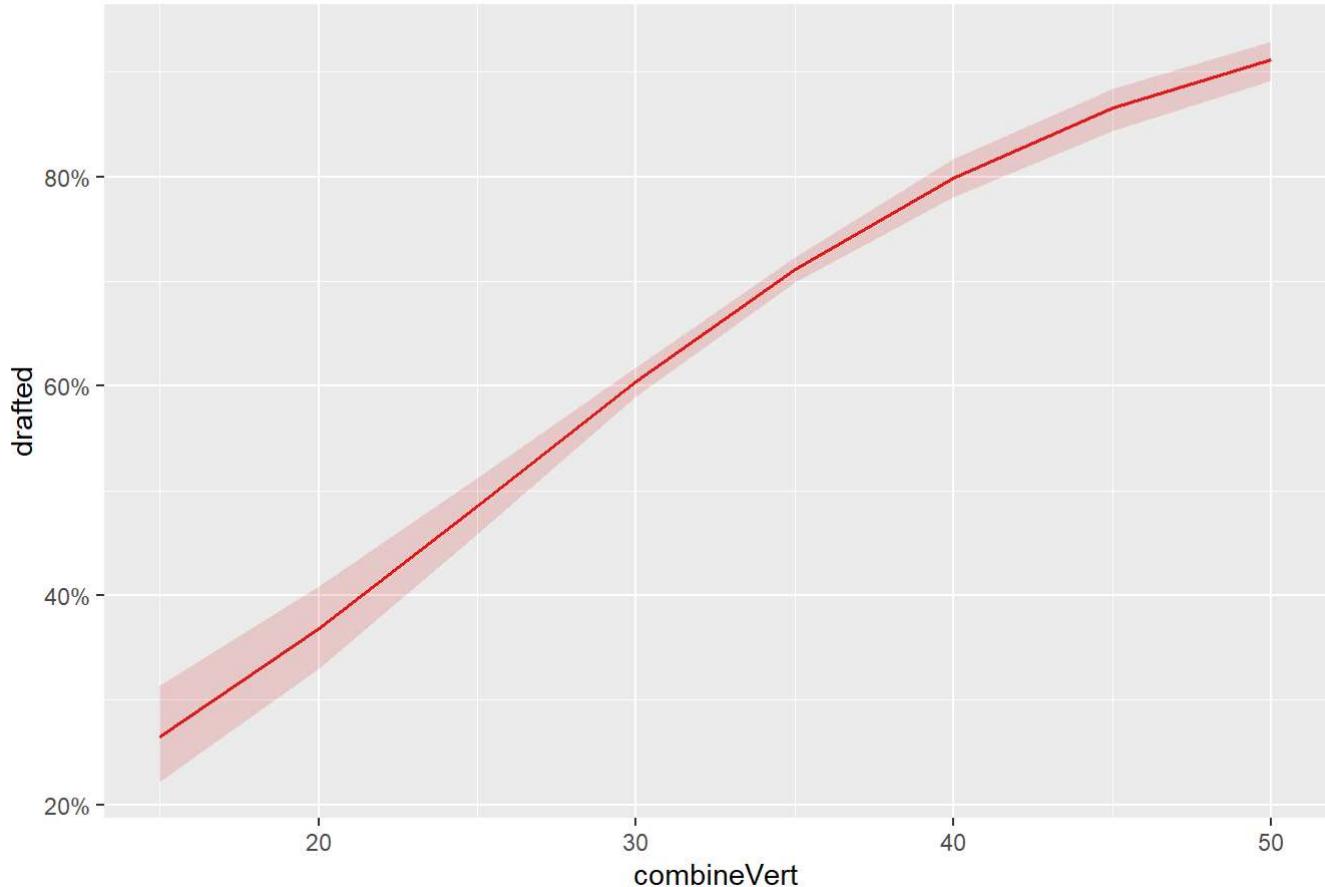
Null deviance: 7572.0 on 5978 degrees of freedom  
 Residual deviance: 7489.8 on 5977 degrees of freedom  
 (2696 observations deleted due to missingness)  
 AIC: 7493.8

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model1_5$terms)[3], AIC(model1_5))
aic_models <- rbind(aic_models, x)
sjPlot::plot_model(model1_2, type = "pred") #part of logistic regression curve
```

Data were 'prettified'. Consider using `terms="combineVert [all]"` to get smooth plots.

## Predicted probabilities of drafted



## 2 Main Effect Terms

```
model2 <- glm(drafted ~ combine60ydShuttle + combineBench, data = combine_draft, family = binomial)
summary(model2)
```

Call:

```
glm(formula = drafted ~ combine60ydShuttle + combineBench, family = binomial,
  data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )							
(Intercept)	14.03080	1.73444	8.090	5.99e-16 ***							
combine60ydShuttle	-1.21301	0.14915	-8.133	4.20e-16 ***							
combineBench	0.05237	0.01121	4.669	3.02e-06 ***							
---											
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'.'	0.1	' '	1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1627.4 on 1225 degrees of freedom

Residual deviance: 1545.3 on 1223 degrees of freedom

(7449 observations deleted due to missingness)

AIC: 1551.3

Number of Fisher Scoring iterations: 4

```
performance::check_collinearity(model2)
```

# Check for Multicollinearity

Low Correlation

Term	VIF	VIF 95% CI	Increased SE	Tolerance	Tolerance 95% CI
combine60ydShuttle	1.06	[1.02, 1.17]	1.03	0.95	[0.86, 0.98]
combineBench	1.06	[1.02, 1.17]	1.03	0.95	[0.86, 0.98]

```
x <- list(as.character(model2$terms)[3], AIC(model2))
aic_models <- rbind(aic_models, x)

exp(model2$coefficients)
```

(Intercept)	combine60ydShuttle	combineBench
1.240222e+06	2.973022e-01	1.053760e+00

```
margins::margins(model2)
```

## Average marginal effects

```
glm(formula = drafted ~ combine60ydShuttle + combineBench, family = binomial,      data =
combine_draft)
```

combine60ydShuttle	combineBench
-0.2668	0.01152

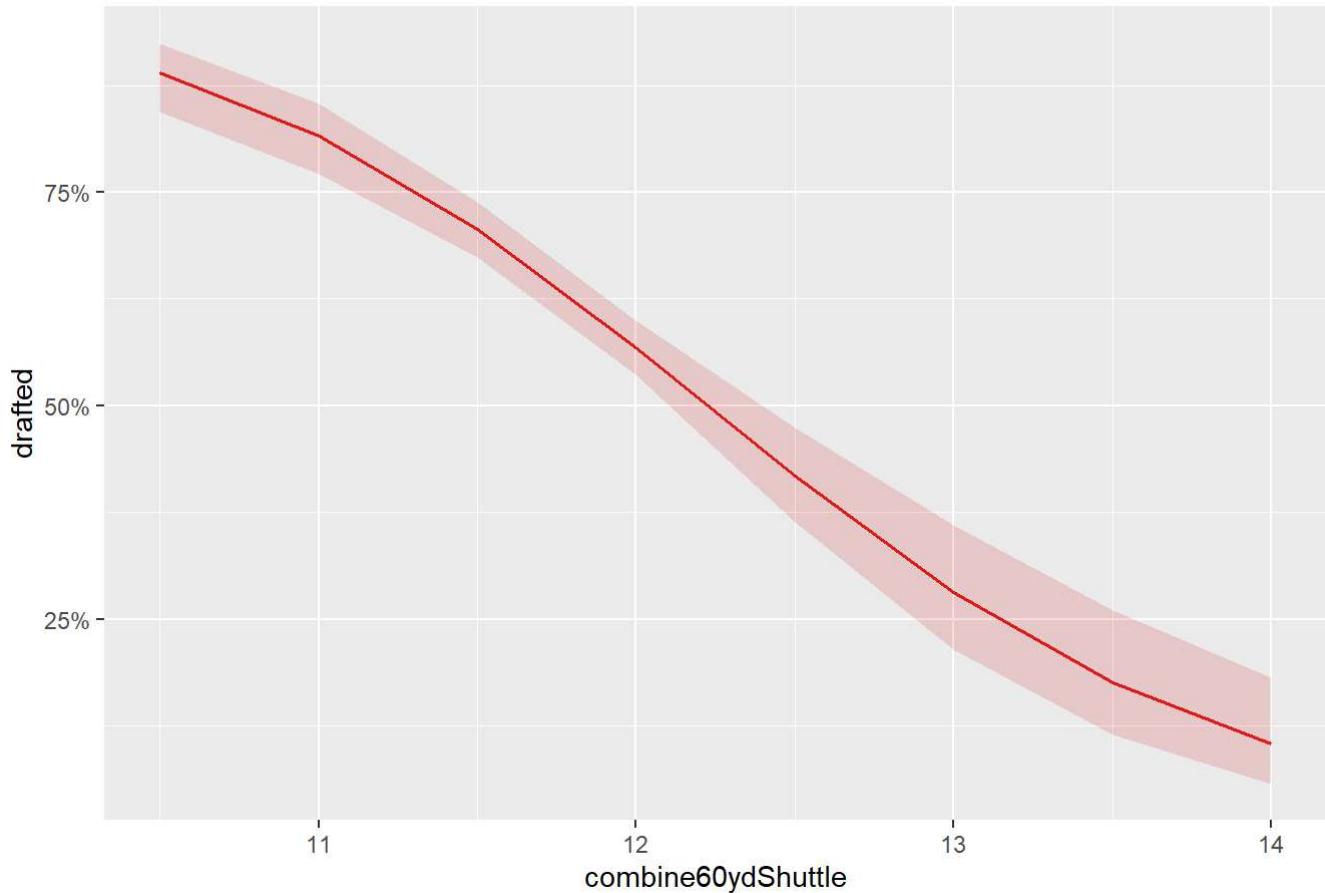
```
sjPlot::plot_model(model2, type ="pred") #part of logistic regression curve
```

Data were 'prettified'. Consider using `terms="combine60ydShuttle [all]"` to get smooth plots.

Data were 'prettified'. Consider using `terms="combineBench [all]"` to get smooth plots.

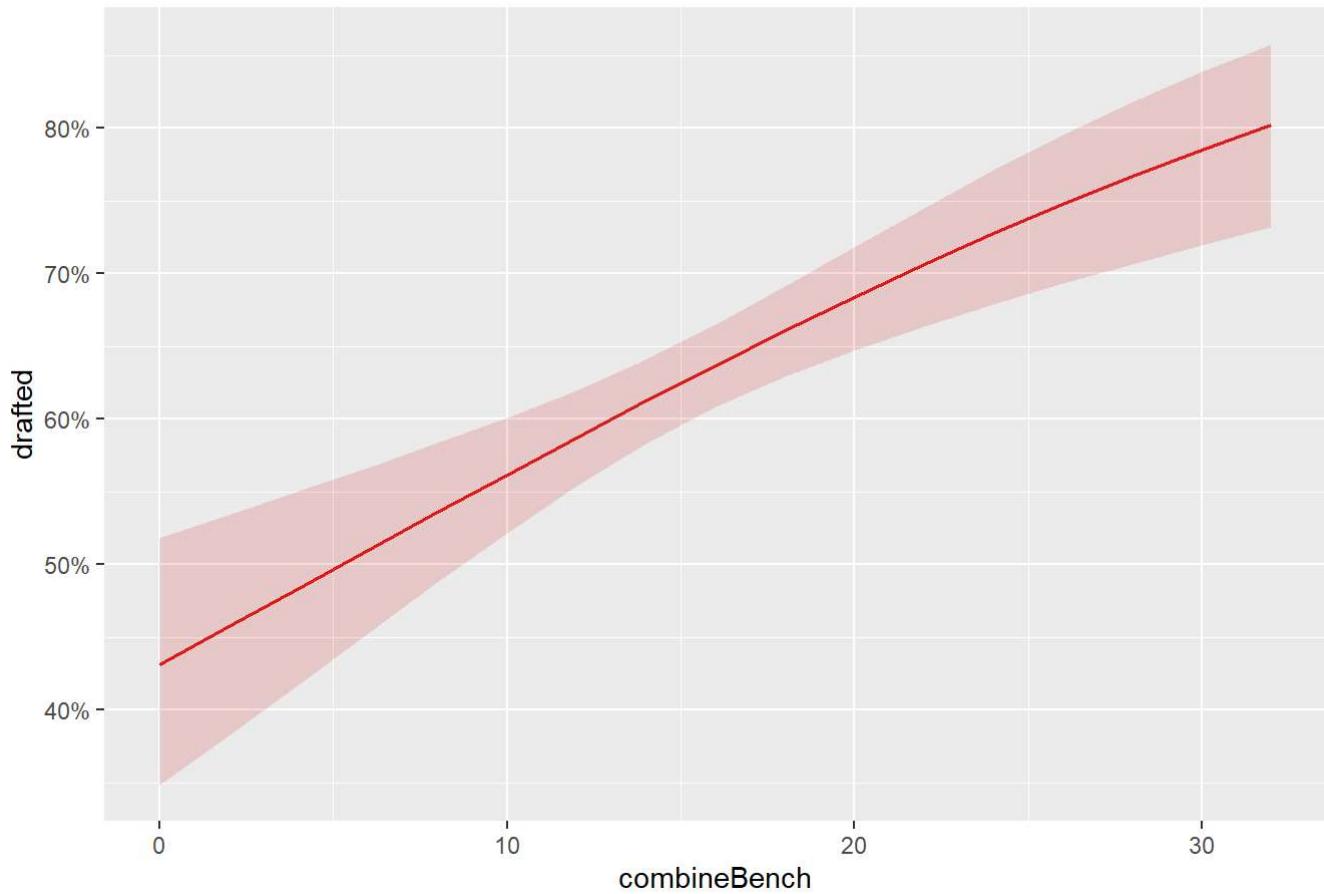
```
$combine60ydShuttle
```

## Predicted probabilities of drafted



```
$combineBench
```

## Predicted probabilities of drafted



For this model, both 60yd shuttle and bench are significant at the 0.01% significance level as their p value < 0.001 and have z values < -1.96 or > 1.96.

Looking at the exponents, you can see for every 1s increase in shuttle your odds of getting drafted decrease (odds < 1) but for bench, for every increase by 1 rep, your odds of getting drafted increase (odds > 1)

Looking at the margins and the logistic curve., on average for every 1 rep increase in bench, the probability of getting drafted increases by 0.012, when all other conditions are constant. On average for every 1 sec increase in 60yd shuttle time, the probability of getting drafted decreases by 0.27, when all other conditions are constant.

```
model2_1 <- glm(drafted ~ combine60ydShuttle + combineBroad, data = combine_draft, family = binomial)
summary(model2_1)
```

Call:

```
glm(formula = drafted ~ combine60ydShuttle + combineBroad, family = binomial,
     data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	1.908772	2.199235	0.868	0.385

```
combine60ydShuttle -0.632954  0.133352 -4.746 2.07e-06 ***
combineBroad       0.052677  0.008922  5.904 3.54e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 2329.9 on 1755 degrees of freedom
Residual deviance: 2223.0 on 1753 degrees of freedom
(6919 observations deleted due to missingness)
AIC: 2229
```

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model2_1$terms)[3], AIC(model2_1))
aic_models <- rbind(aic_models, x)
performance::check_collinearity(model2_1)
```

# Check for Multicollinearity

Low Correlation

	Term	VIF	VIF 95% CI	Increased SE	Tolerance	Tolerance 95% CI
combine60ydShuttle	1.21	[1.15, 1.29]		1.10	0.83	[0.78, 0.87]
combineBroad	1.21	[1.15, 1.29]		1.10	0.83	[0.78, 0.87]

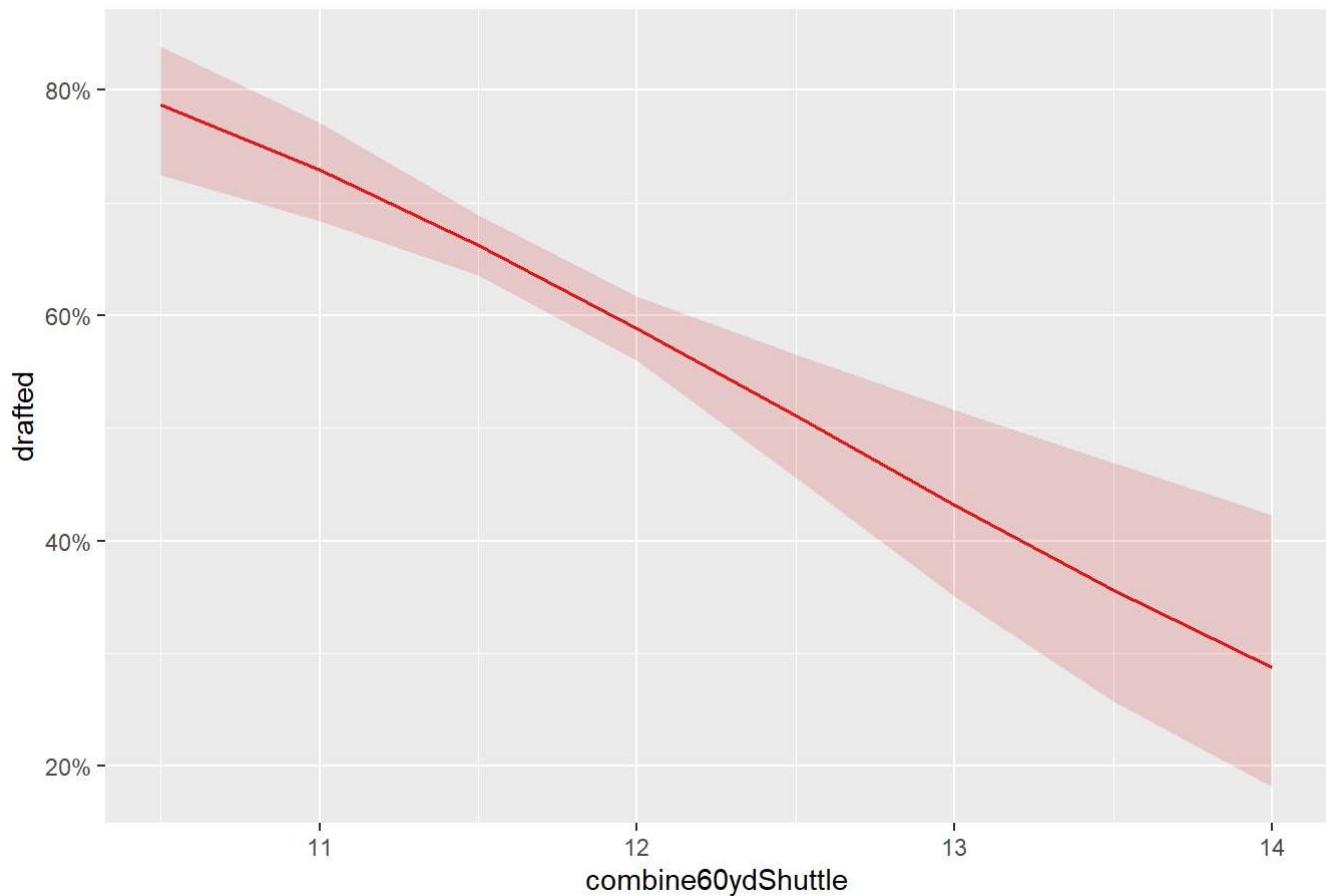
```
sjPlot::plot_model(model2_1, type = "pred")
```

Data were 'prettified'. Consider using `terms="combine60ydShuttle [all]"` to get smooth plots.

Data were 'prettified'. Consider using `terms="combineBroad [all]"` to get smooth plots.

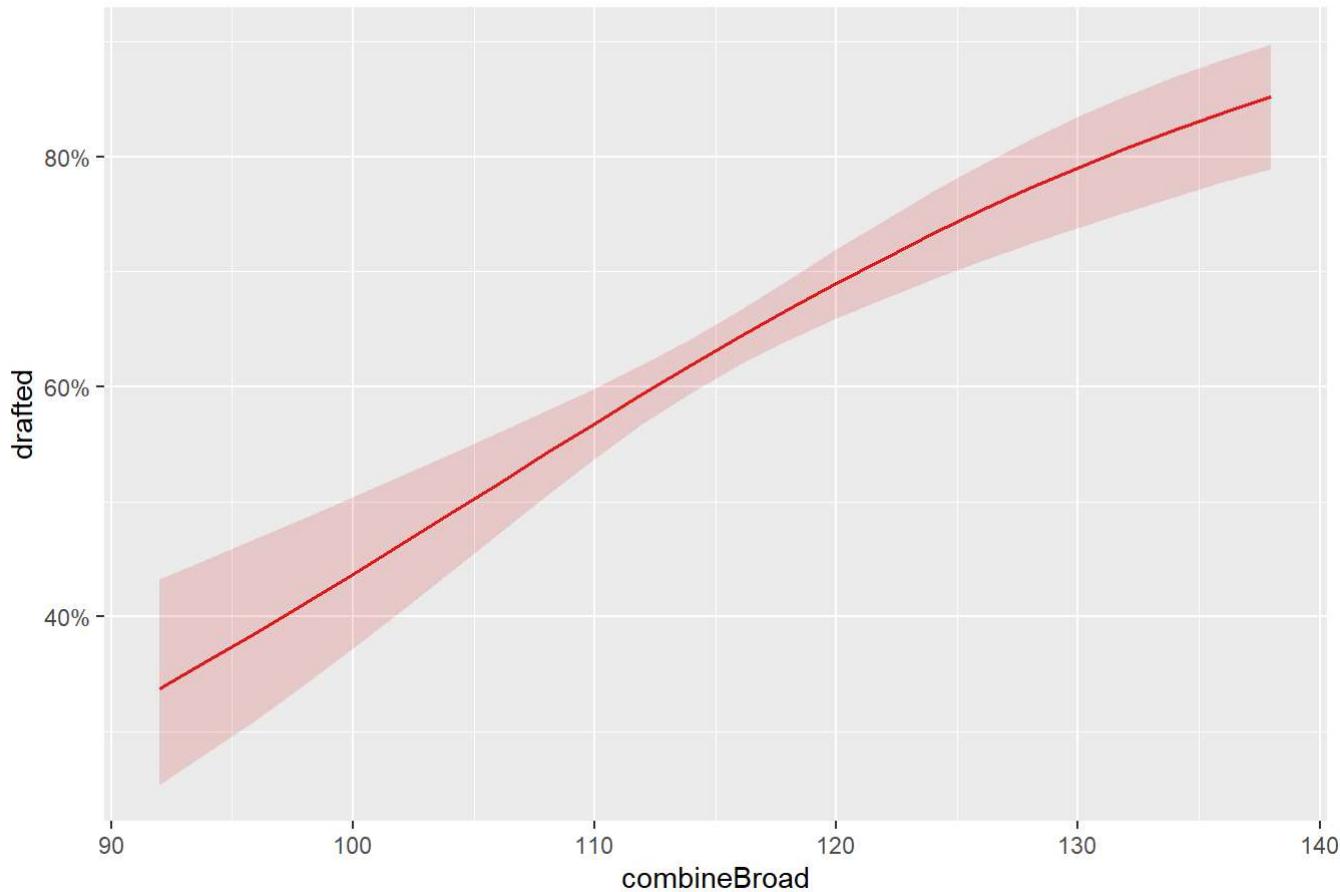
\$combine60ydShuttle

### Predicted probabilities of drafted



\$combineBroad

### Predicted probabilities of drafted



## 2 Interaction Terms

```
model2_2 <- glm(drafted ~ combine60ydShuttle * combineBench, data = combine_draft, family = binomial)
summary(model2_2)
```

Call:

```
glm(formula = drafted ~ combine60ydShuttle * combineBench, family = binomial,
     data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	19.94830	5.42411	3.678	0.000235 ***
combine60ydShuttle	-1.71525	0.46103	-3.720	0.000199 ***
combineBench	-0.32773	0.32833	-0.998	0.318201
combine60ydShuttle:combineBench	0.03217	0.02778	1.158	0.246864
---				
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 ' '	1		

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 1627.4 on 1225 degrees of freedom
Residual deviance: 1544.0 on 1222 degrees of freedom
(7449 observations deleted due to missingness)
AIC: 1552
```

Number of Fisher Scoring iterations: 4

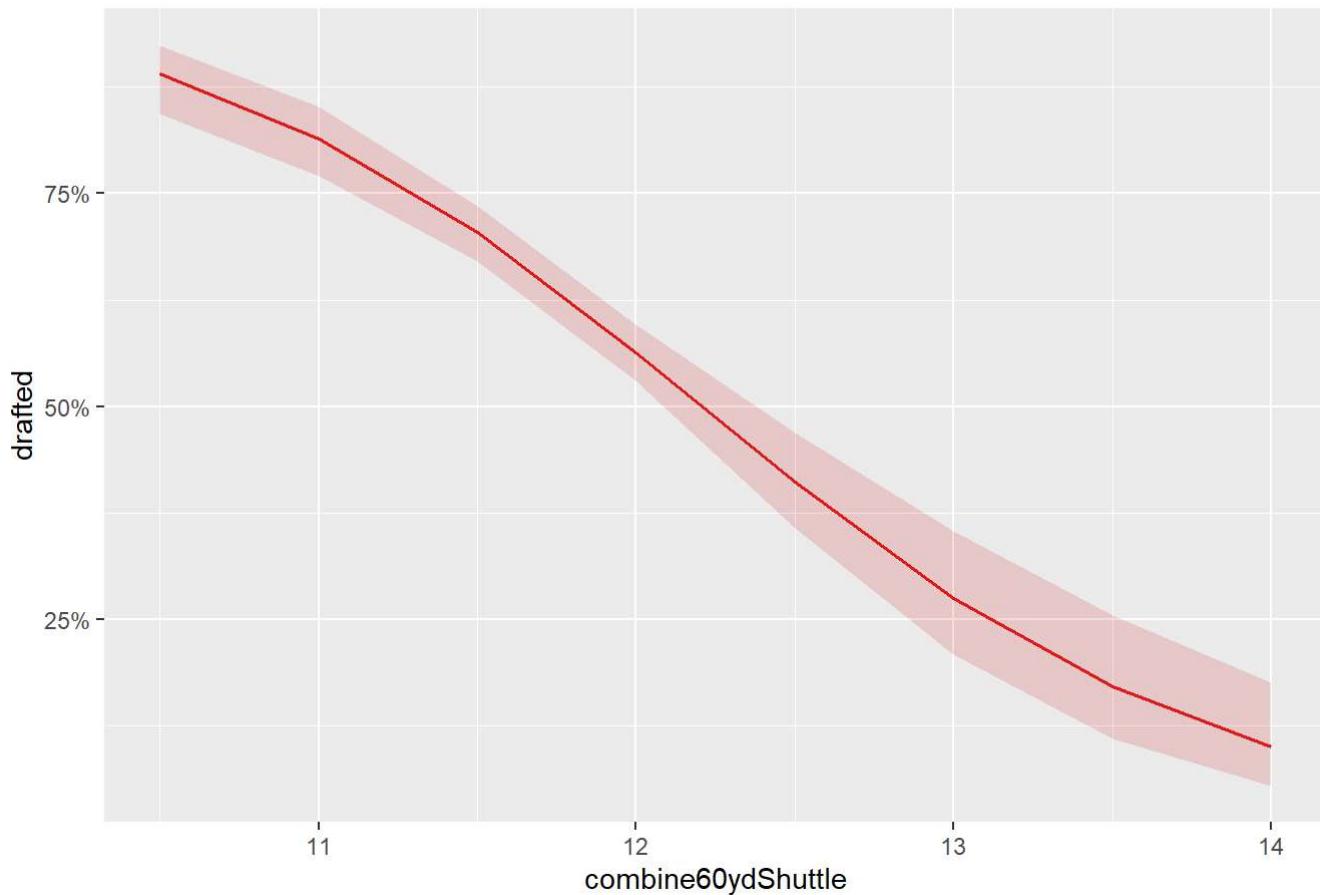
```
x <- list(as.character(model2_2$terms)[3], AIC(model2_2))
aic_models <- rbind(aic_models, x)
sjPlot::plot_model(model2_2, type = "pred")
```

Data were 'prettified'. Consider using `terms="combine60ydShuttle [all]"` to get smooth plots.

Data were 'prettified'. Consider using `terms="combineBench [all]"` to get smooth plots.

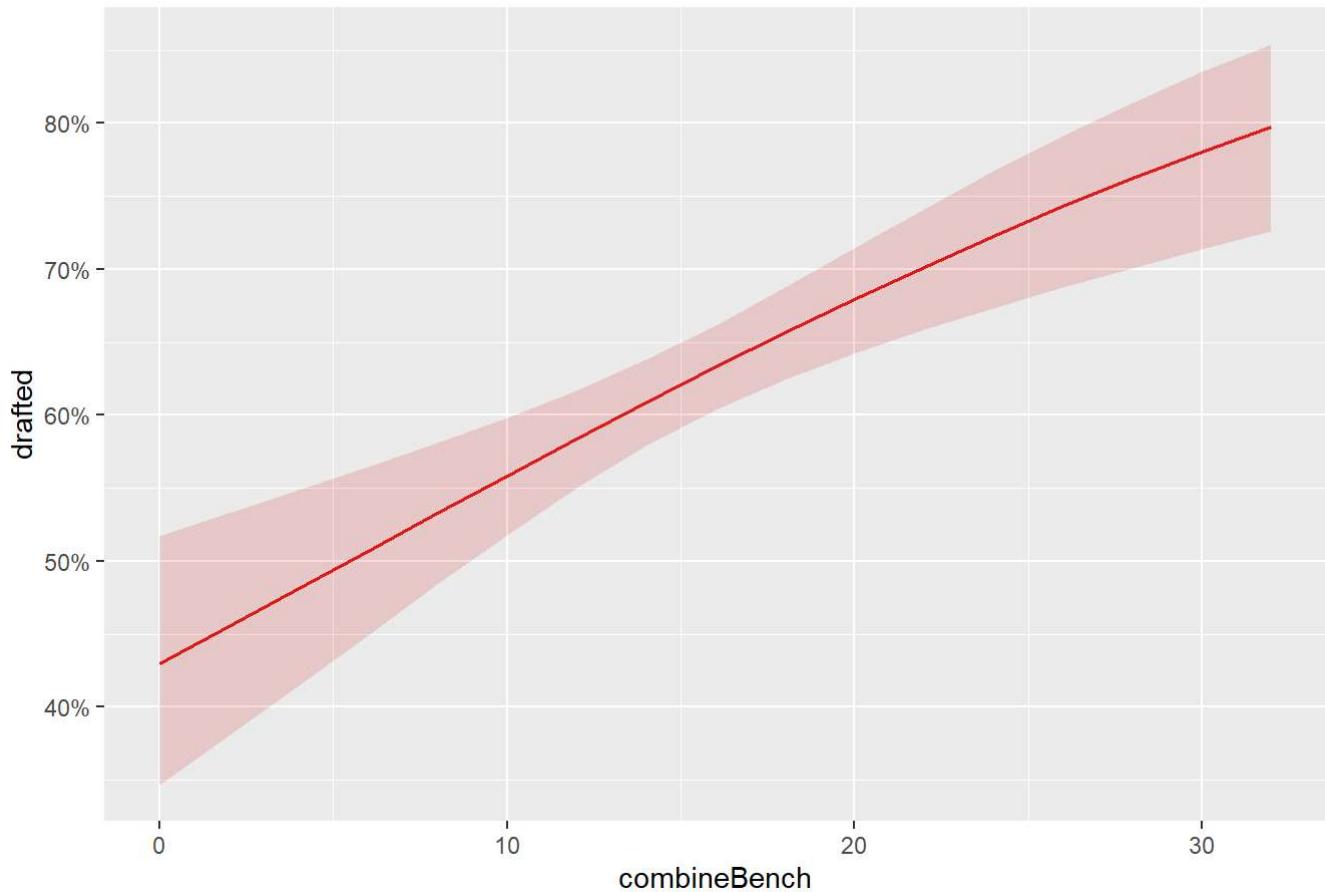
\$combine60ydShuttle

### Predicted probabilities of drafted



\$combineBench

## Predicted probabilities of drafted



```
model12_3 <- glm(drafted ~ combine60ydShuttle * combineBroad, data = combine_draft, family = binomial)
summary(model12_3)
```

Call:

```
glm(formula = drafted ~ combine60ydShuttle * combineBroad, family = binomial,
     data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-50.12283	22.89521	-2.189	0.0286 *
combine60ydShuttle	3.77385	1.93234	1.953	0.0508 .
combineBroad	0.51321	0.20228	2.537	0.0112 *
combine60ydShuttle:combineBroad	-0.03904	0.01711	-2.282	0.0225 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 2329.9 on 1755 degrees of freedom

Residual deviance: 2217.8 on 1752 degrees of freedom

(6919 observations deleted due to missingness)

AIC: 2225.8

Number of Fisher Scoring iterations: 4

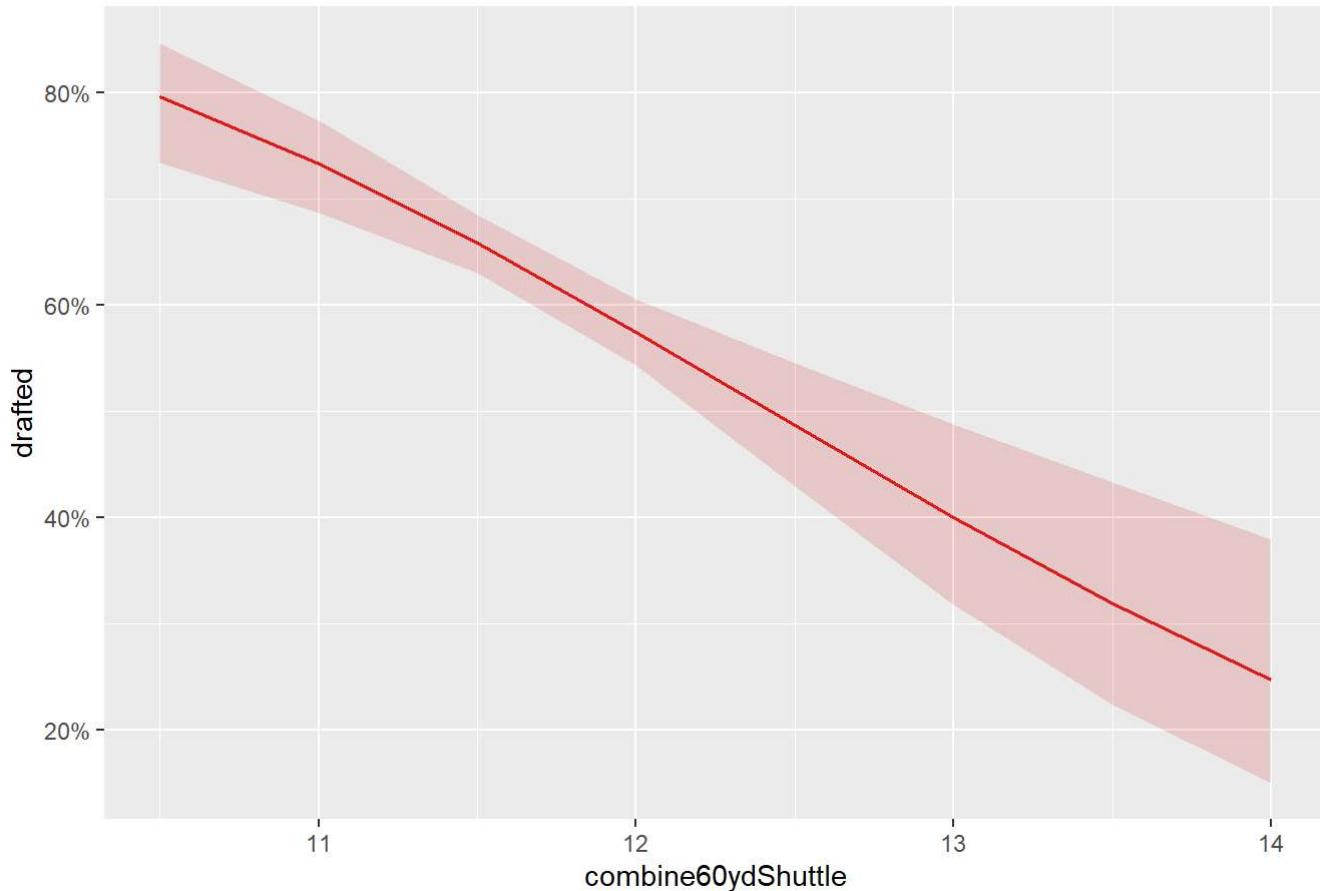
```
x <- list(as.character(model2_3$terms)[3], AIC(model2_3))
aic_models <- rbind(aic_models, x)
sjPlot::plot_model(model2_3, type = "pred")
```

Data were 'prettified'. Consider using `terms="combine60ydShuttle [all]"` to get smooth plots.

Data were 'prettified'. Consider using `terms="combineBroad [all]"` to get smooth plots.

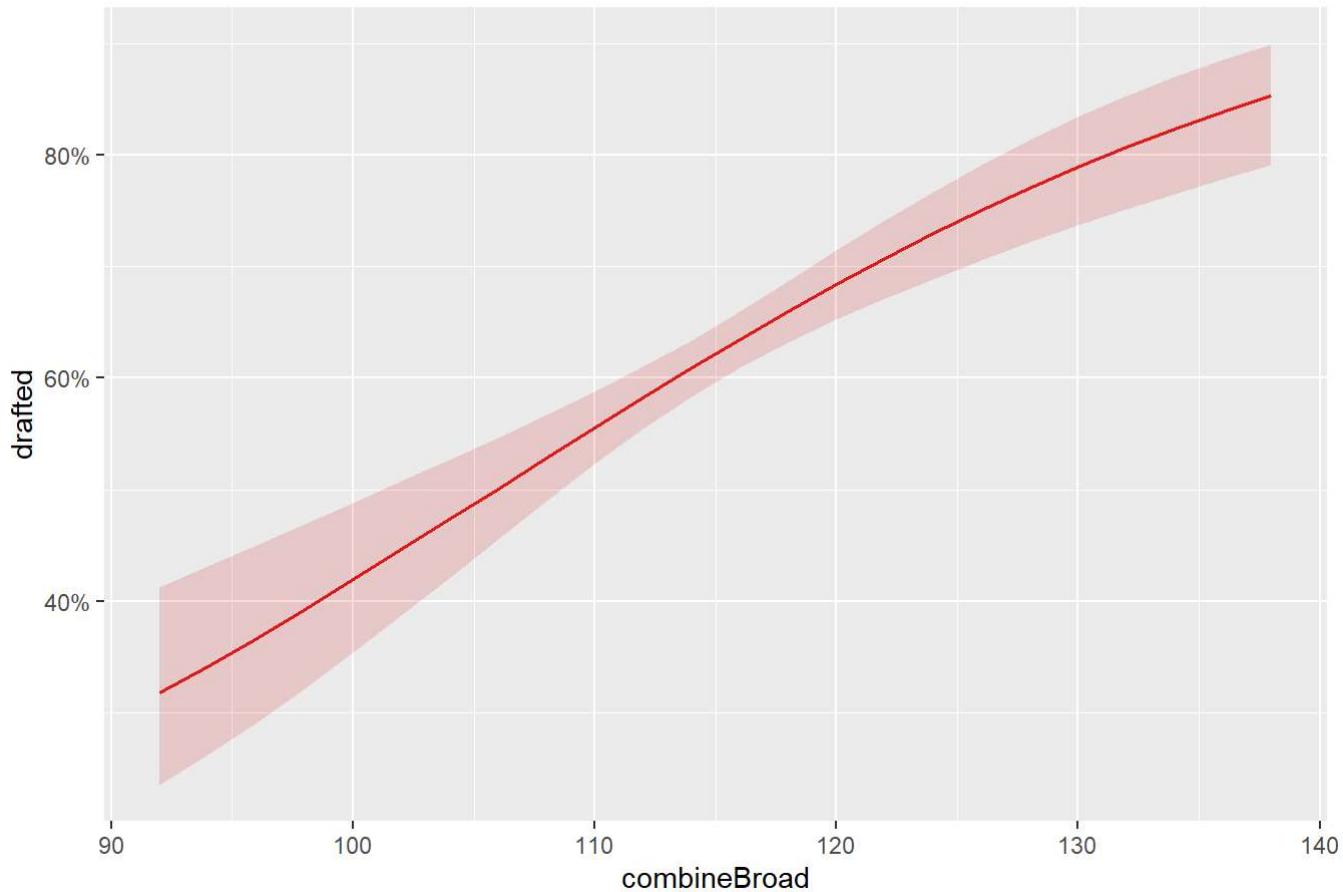
```
$combine60ydShuttle
```

### Predicted probabilities of drafted



```
$combineBroad
```

### Predicted probabilities of drafted



### 3 Main Effect Terms

```
model3 <- glm(drafted ~ combineBroad + combineVert + combine60ydShuttle, data = combine_draft)
summary(model3)
```

Call:

```
glm(formula = drafted ~ combineBroad + combineVert + combine60ydShuttle,
family = binomial, data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	1.28685	2.21675	0.581	0.56157
combineBroad	0.03005	0.01170	2.568	0.01024 *
combineVert	0.07271	0.02516	2.890	0.00385 **
combine60ydShuttle	-0.56021	0.13567	-4.129	3.64e-05 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

```
Null deviance: 2322.2 on 1750 degrees of freedom
Residual deviance: 2208.8 on 1747 degrees of freedom
(6924 observations deleted due to missingness)
AIC: 2216.8
```

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model3$terms)[3], AIC(model3))
aic_models <- rbind(aic_models, x)
performance::check_collinearity(model3)
```

# Check for Multicollinearity

Low Correlation

	Term	VIF	VIF 95% CI	Increased SE	Tolerance	Tolerance 95% CI
	combineBroad	2.07	[1.93, 2.23]	1.44	0.48	[0.45, 0.52]
	combineVert	2.04	[1.91, 2.20]	1.43	0.49	[0.45, 0.52]
	combine60ydShuttle	1.25	[1.19, 1.33]	1.12	0.80	[0.75, 0.84]

```
sjPlot::plot_model(model3, type = "pred") #part of logistic regression curve
```

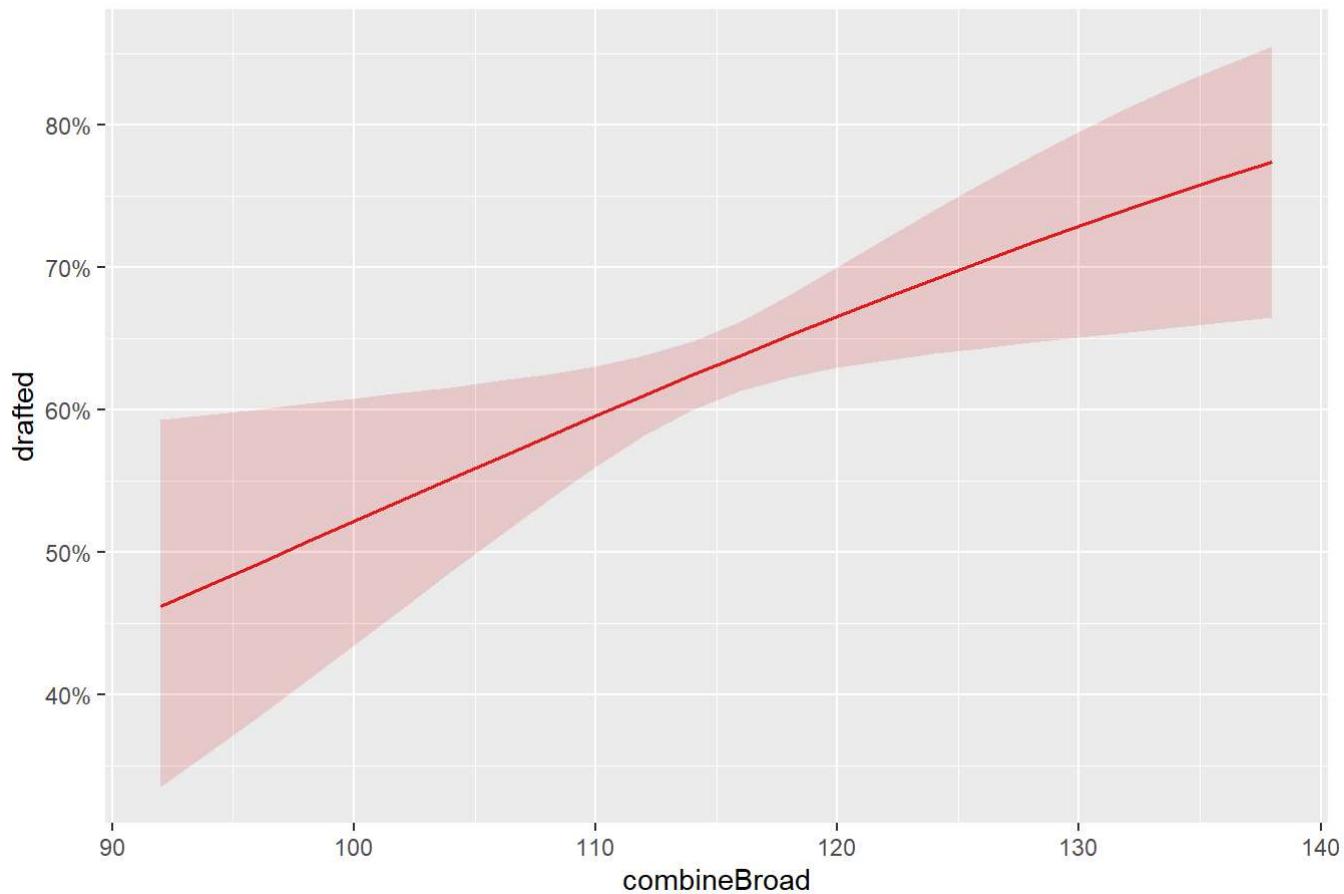
Data were 'prettified'. Consider using `terms="combineBroad [all]"` to get smooth plots.

Data were 'prettified'. Consider using `terms="combineVert [all]"` to get smooth plots.

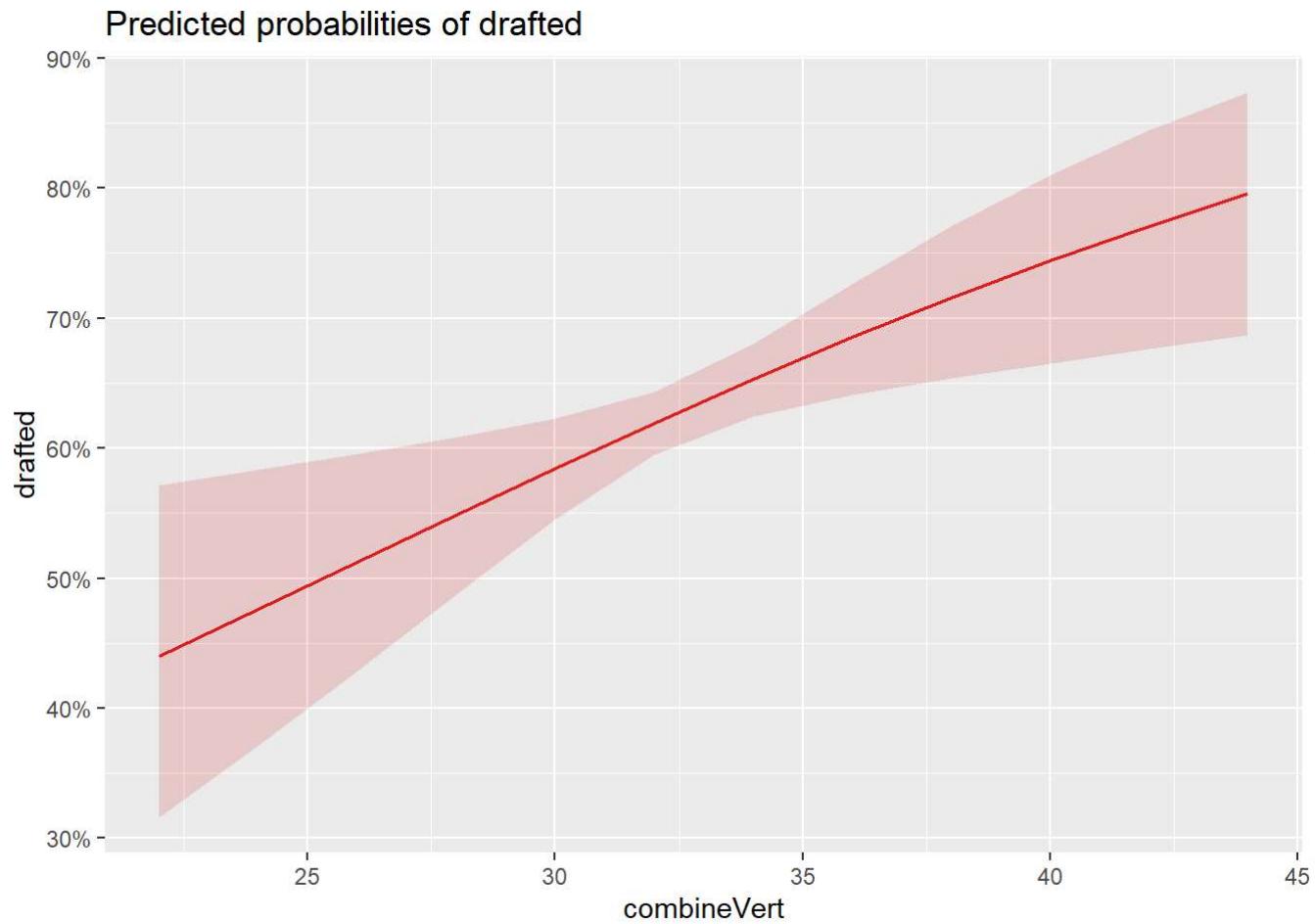
Data were 'prettified'. Consider using `terms="combine60ydShuttle [all]"` to get smooth plots.

\$combineBroad

### Predicted probabilities of drafted

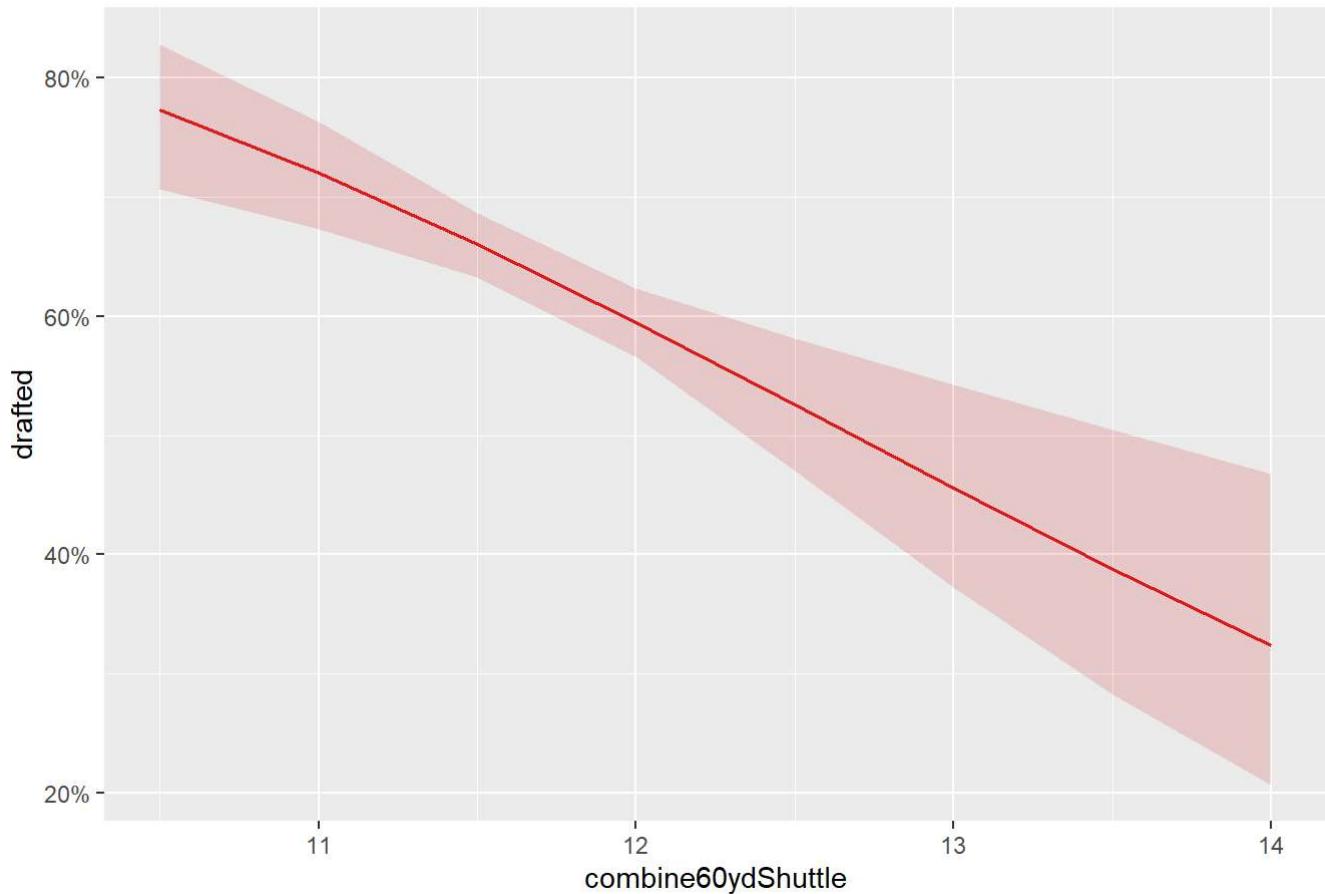


\$combineVert



\$combine60ydShuttle

## Predicted probabilities of drafted



```
model3_1 <- glm(drafted ~ combineBroad + combineBench + combine60ydShuttle, data = combine_draft)
summary(model3_1)
```

Call:

```
glm(formula = drafted ~ combineBroad + combineBench + combine60ydShuttle,
family = binomial, data = combine_draft)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	7.48535	2.66880	2.805	0.00504 **
combineBroad	0.03412	0.01063	3.210	0.00133 **
combineBench	0.05078	0.01131	4.488	7.18e-06 ***
combine60ydShuttle	-0.98592	0.16525	-5.966	2.43e-09 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 1616.8 on 1217 degrees of freedom

Residual deviance: 1524.5 on 1214 degrees of freedom

(7457 observations deleted due to missingness)

AIC: 1532.5

Number of Fisher Scoring iterations: 4

```
x <- list(as.character(model3_1$terms)[3], AIC(model3_1))
aic_models <- rbind(aic_models, x)
performance::check_collinearity(model3_1)
```

# Check for Multicollinearity

Low Correlation

Term	VIF	VIF 95% CI	Increased SE	Tolerance	Tolerance 95% CI
combineBroad	1.22	[1.15, 1.31]	1.10	0.82	[0.76, 0.87]
combineBench	1.06	[1.02, 1.17]	1.03	0.95	[0.85, 0.98]
combine60ydShuttle	1.28	[1.20, 1.38]	1.13	0.78	[0.73, 0.83]

```
sjPlot::plot_model(model3_1, type = "pred") #part of logistic regression curve
```

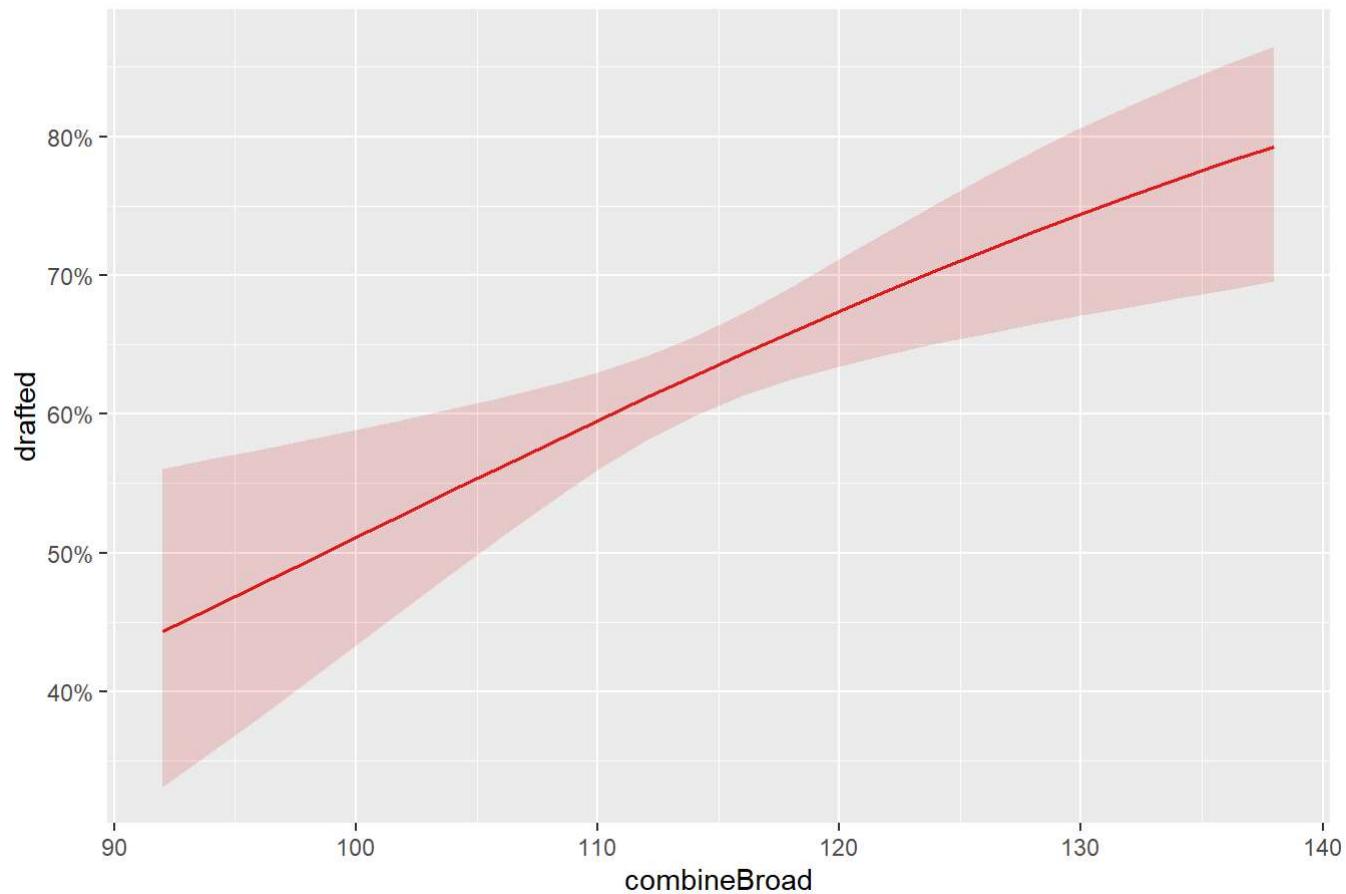
Data were 'prettified'. Consider using `terms="combineBroad [all]"` to get smooth plots.

Data were 'prettified'. Consider using `terms="combineBench [all]"` to get smooth plots.

Data were 'prettified'. Consider using `terms="combine60ydShuttle [all]"` to get smooth plots.

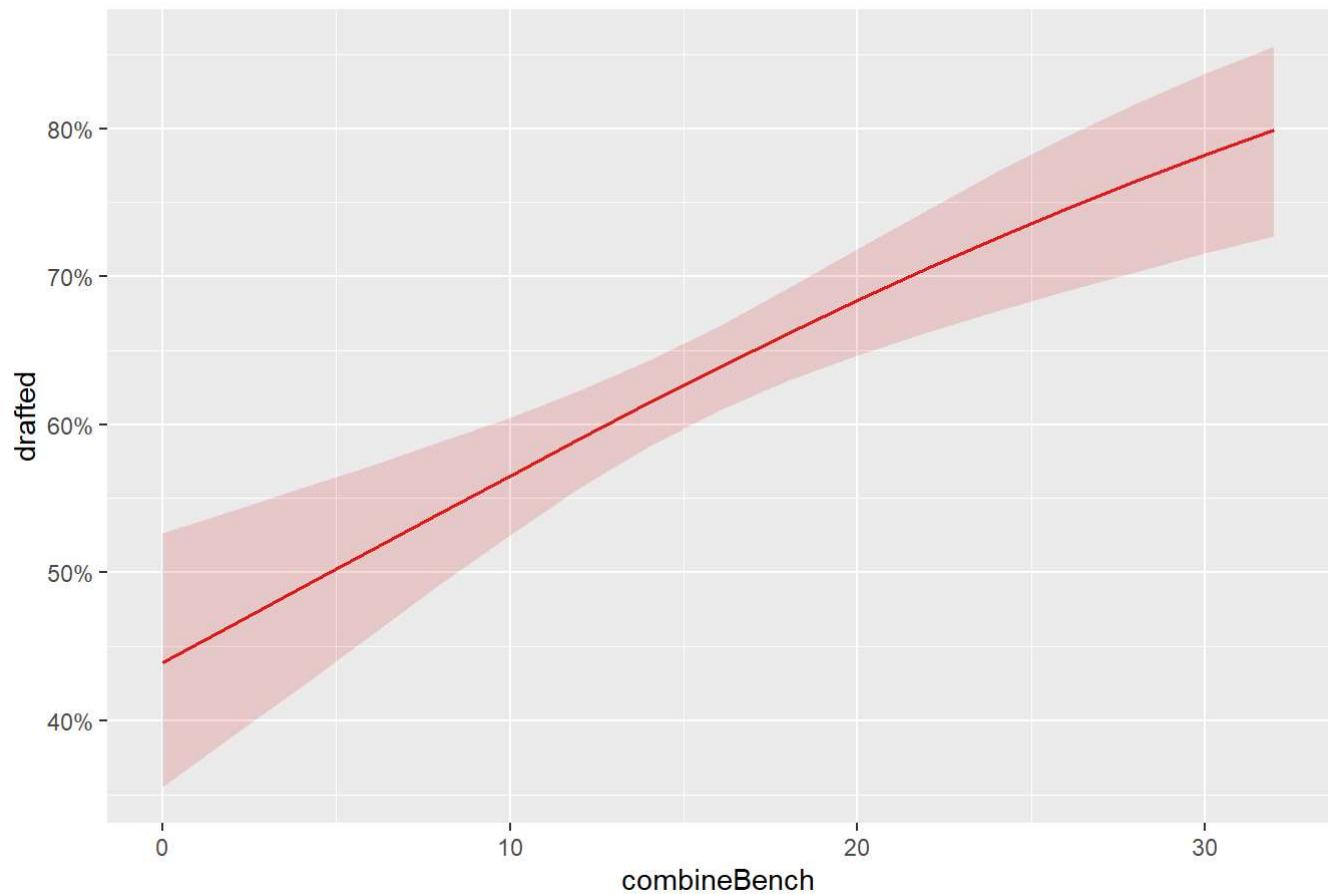
\$combineBroad

### Predicted probabilities of drafted



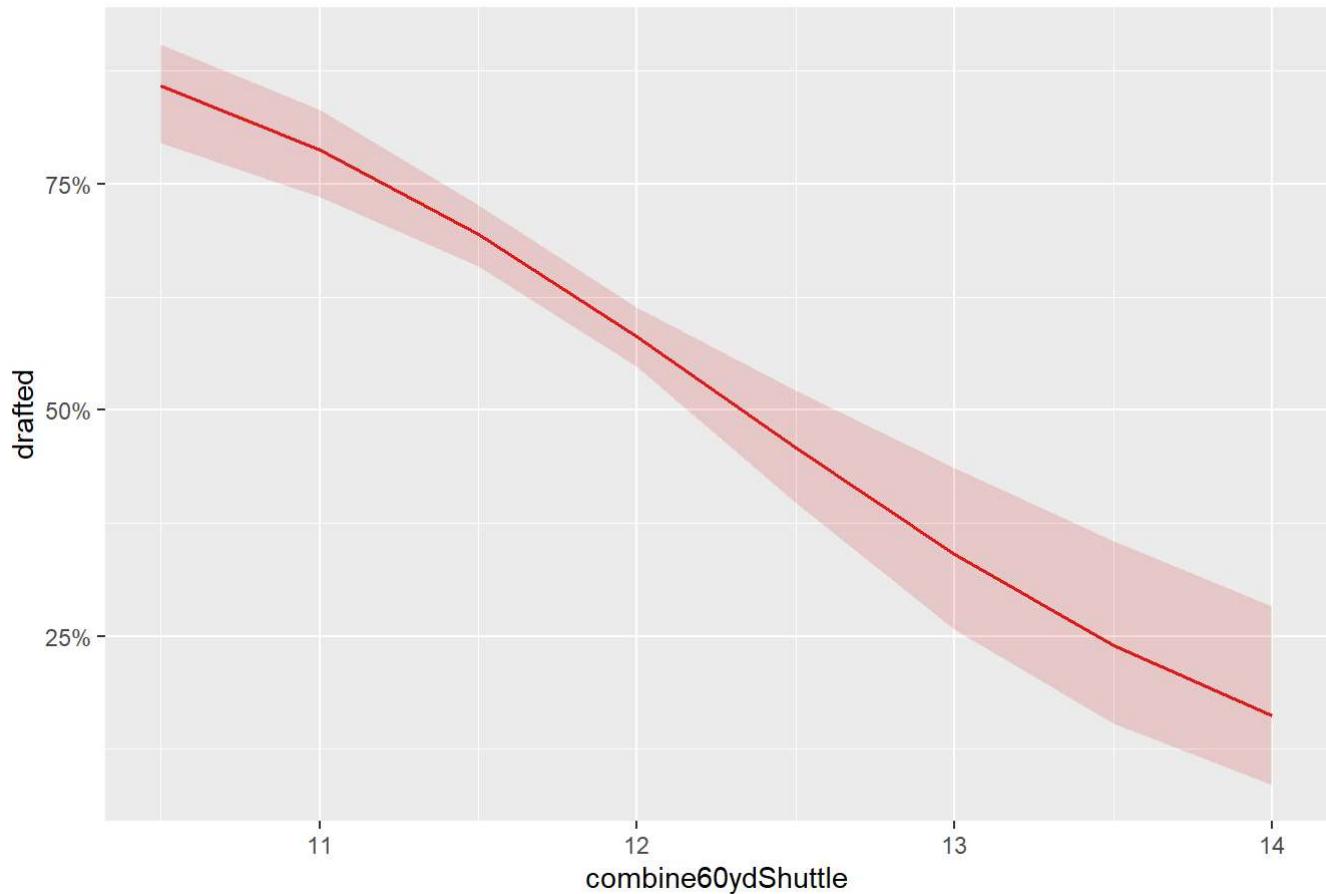
\$combineBench

### Predicted probabilities of drafted



\$combine60ydShuttle

## Predicted probabilities of drafted



## Model for part 2

### Generalised Linear Model

```
weight1 <- lm(pick ~ combineWeight, data = combine_draft)
summary(weight1)
```

Call:

```
lm(formula = pick ~ combineWeight, data = combine_draft)
```

Residuals:

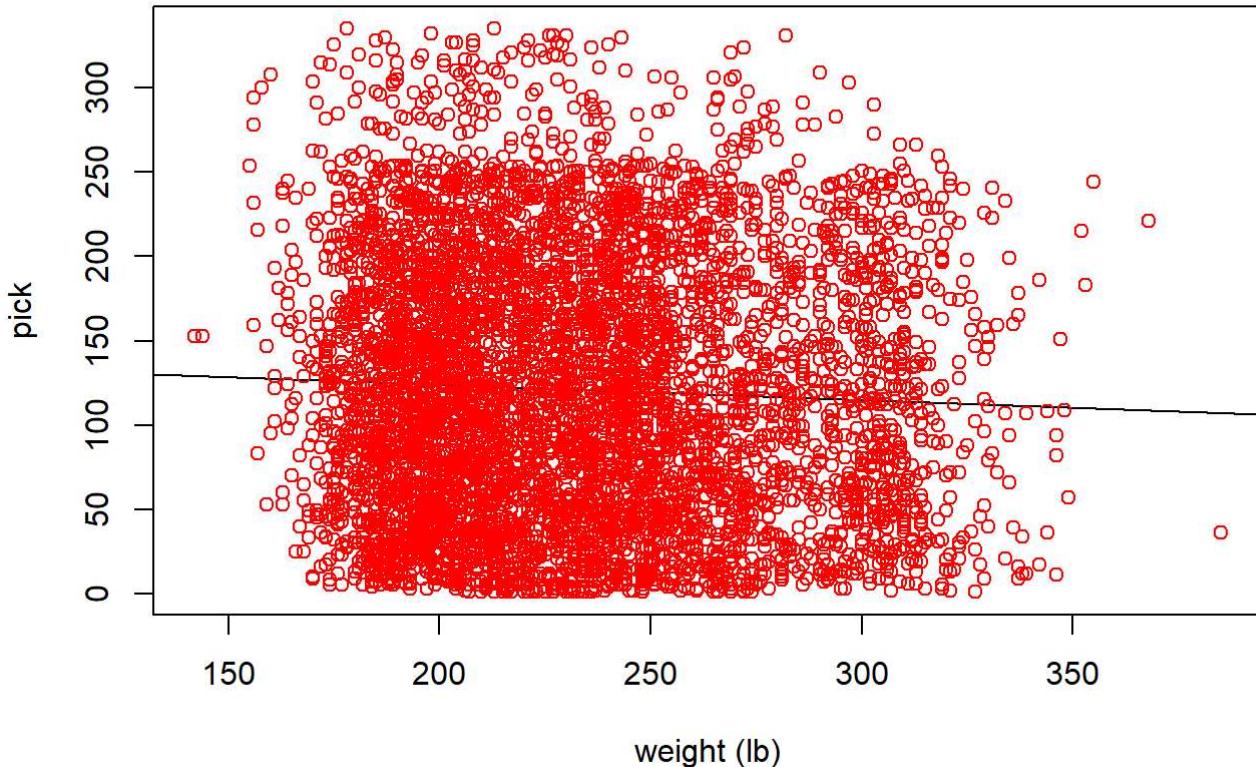
Min	1Q	Median	3Q	Max
-122.313	-64.109	-7.297	58.097	214.450

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	141.97943	6.08016	23.351	< 2e-16 ***
combineWeight	-0.09018	0.02588	-3.484	0.000497 ***
---				
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 ' '	1		

Residual standard error: 75.99 on 5723 degrees of freedom  
 (2950 observations deleted due to missingness)  
 Multiple R-squared: 0.002117, Adjusted R-squared: 0.001943  
 F-statistic: 12.14 on 1 and 5723 DF, p-value: 0.0004969

```
plot(x=combine_draft$combineWeight,y=combine_draft$pick, col = "red"
      , abline(weight1), xlab = "weight (lb)", ylab = "pick")
```



First, I just looked at a generalised linear model to see the effect of weight on pick number. My results show significant evidence at the 0.01% significance level to suggest a correlation between combine weight and pick number. Although, according to my  $r^2$  value 0.002 it is a very weak correlation. The coefficient shows for every 1lb increase in weight, your pick number will decrease by 0.09 (earlier in the draft)

```
ggplot(combine_draft, aes(x = combineWeight, y = pick)) +
  geom_point() +
  facet_wrap(~fullPos) +
  stat_smooth(method = "lm")
```

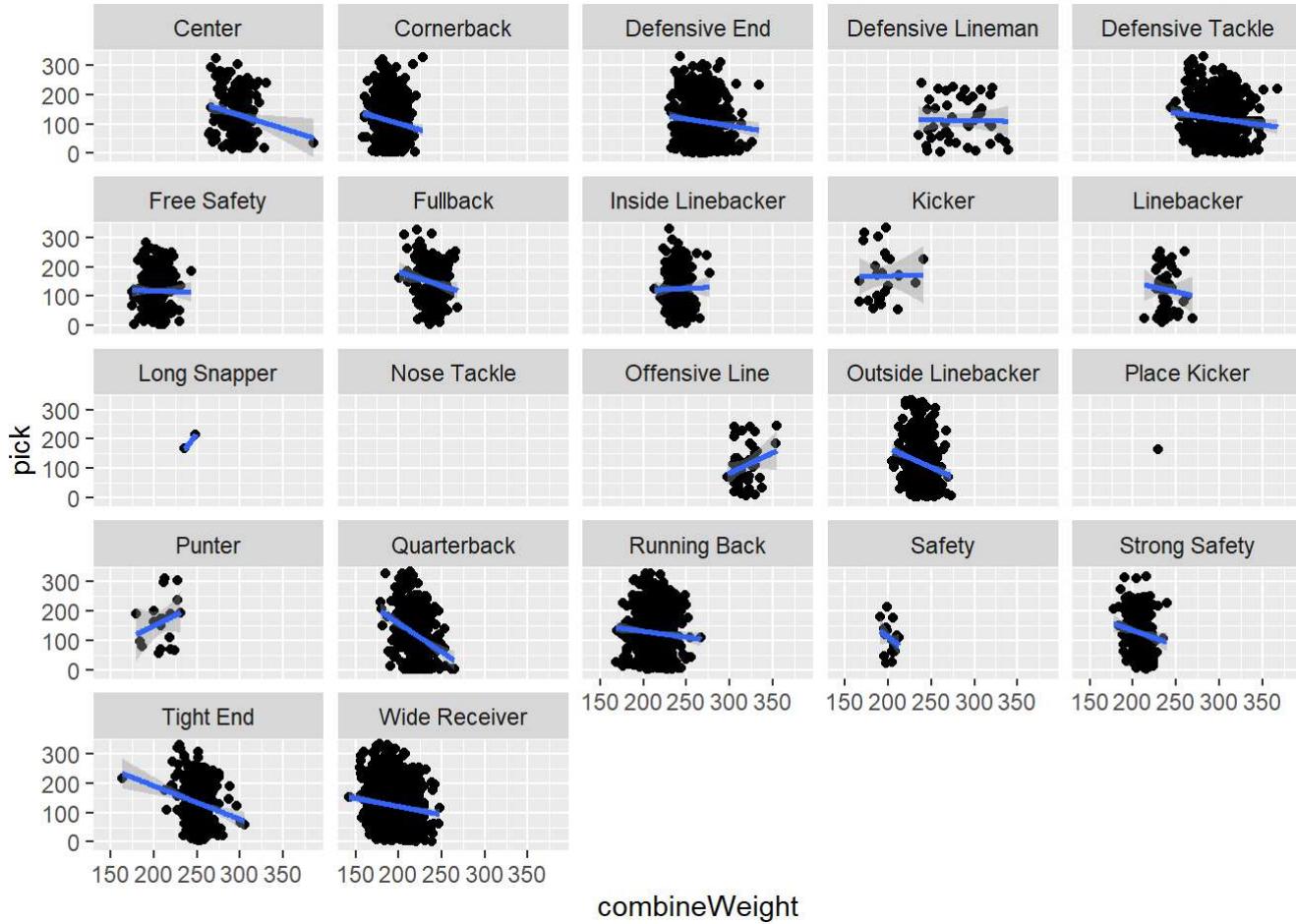
```
`geom_smooth()` using formula = 'y ~ x'
```

Warning: Removed 2950 rows containing non-finite outside the scale range  
 (`stat\_smooth()`).

Warning in qt((1 - level)/2, df): NaNs produced

Warning: Removed 2950 rows containing missing values or values outside the scale range (`geom\_point()`).

Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -Inf



I plotted weight-pick by position to see if there was any difference between positions and if separating them may produce a better model. From my graphs, you can see that different positions produce groupings at different x values. Therefore, I thought it would be worth investigating a mixed effects model.

## Mixed Effects Model

```
weight <- lmer(pick ~ combineWeight + (1|fullPos), data = combine_draft)
summary(weight)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: pick ~ combineWeight + (1 | fullPos)
Data: combine_draft
```

REML criterion at convergence: 65743.1

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.9128	-0.8526	-0.1009	0.7625	3.1445

Random effects:

Groups	Name	Variance	Std.Dev.
fullPos	(Intercept)	465.9	21.58
	Residual	5636.1	75.07

Number of obs: 5725, groups: fullPos, 21

Fixed effects:

	Estimate	Std. Error	t value
(Intercept)	250.31993	15.78044	15.863
combineWeight	-0.52504	0.06264	-8.382

Correlation of Fixed Effects:

	(Intr)
combineWght	-0.944

```
performance::r2(weight)
```

# R2 for Mixed Models

Conditional R2: 0.135

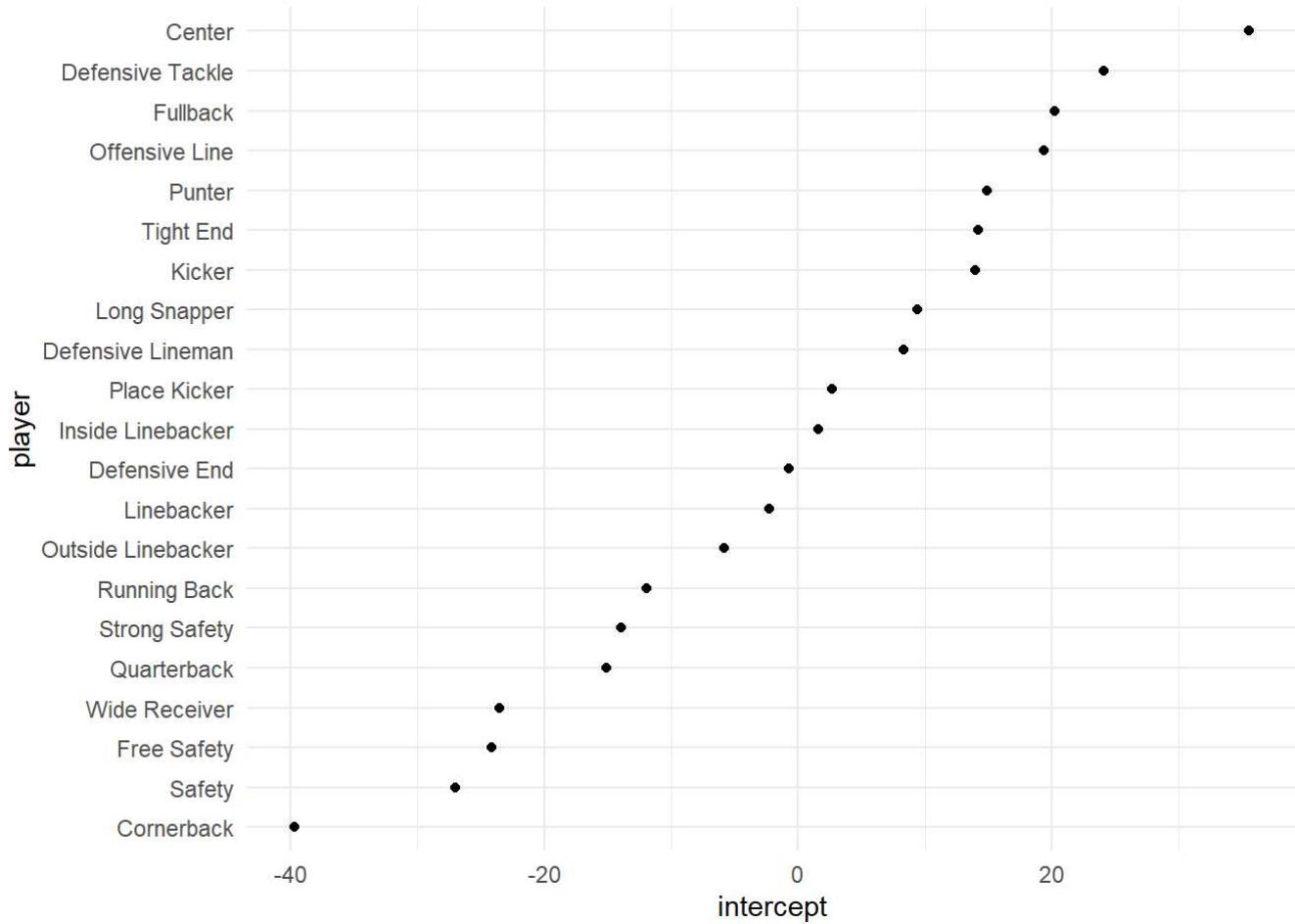
Marginal R2: 0.064

Using the standard fixed effects model, which has a significant t value of -8.4, for every 1lb heavier you are, your pick number will be 0.53 less (earlier in the draft). This indicates that weight may be a significant factor for getting drafted but that it varies by position

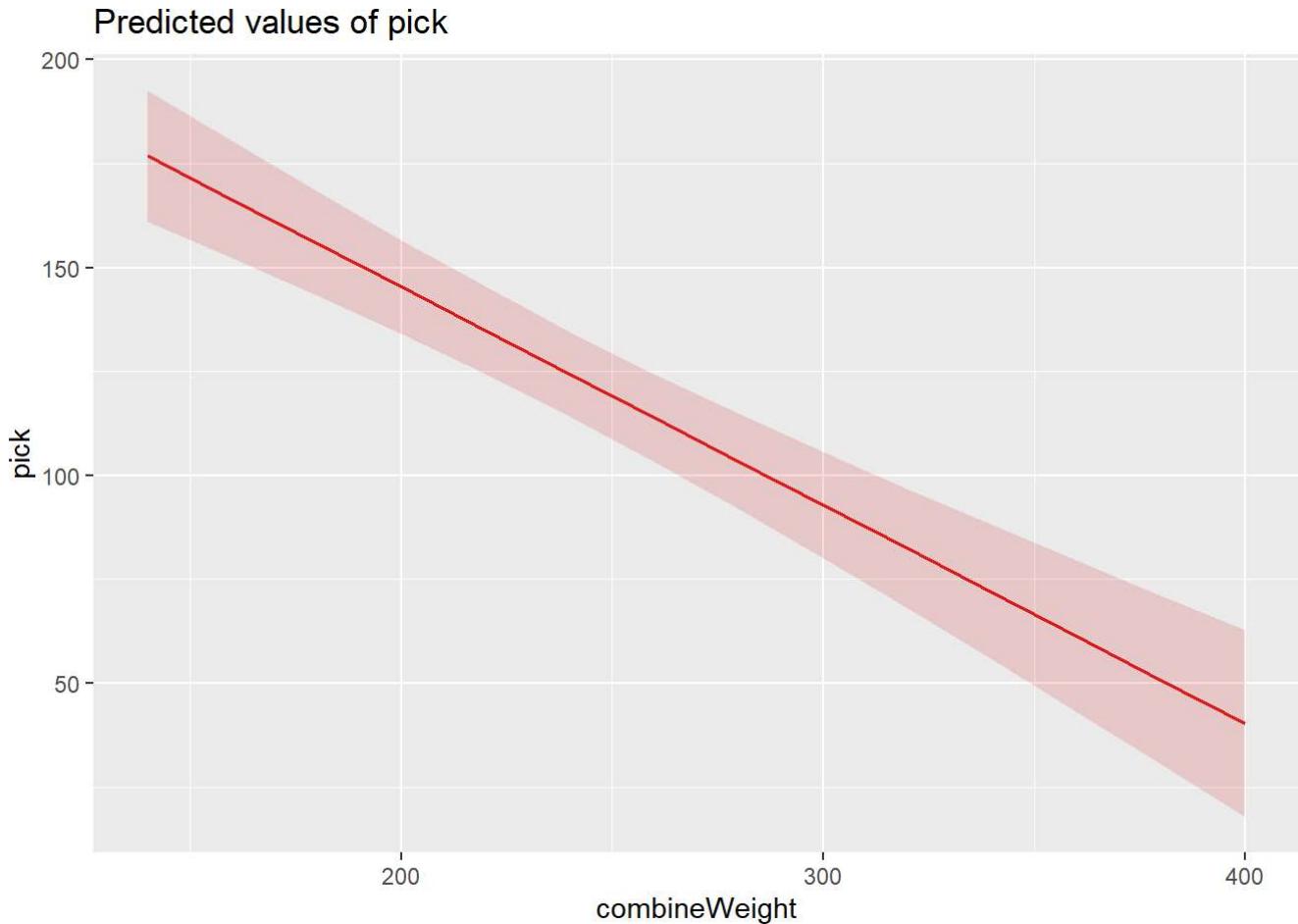
Based on the conditional and marginal R2 values, by adding intercepts by position, this model can account for an additional 7% variance.

```
random_effects = ranef(weight)
ranef_df <- data.frame(
  int = random_effects$fullPos`$(Intercept)`,
  player = rownames(random_effects$fullPos)
)

ggplot(ranef_df, aes(x=int, y=reorder(player, int))) +
  geom_point() +
  labs(x= "intercept", y="player") +
  theme_minimal()
```



```
sjPlot::plot_model(weight, type = "pred") #fixed effects
```



When plotting the intercepts by position we can see that the “starting points” of weight by position differ a lot. As expected the d-line and o-line positions, have a higher intercept when it comes to their linear model, indicating that their weight has a higher baseline. However the generally more athletic positions such as cornerback, safety and receiver, have a much lower intercept indicating that their linear models have a much lower baseline.

**pick ~ combineWeight + (1|fullPos)**

## AIC Comparison

```
aic_models %>% arrange(aic)
```

	model	aic
1	combineBroad + combineBench + combine60ydShuttle	1532.537
2	combine60ydShuttle + combineBench	1551.342
3	combine60ydShuttle * combineBench	1551.991
4	combineBroad + combineVert + combine60ydShuttle	2216.767
5	combine60ydShuttle * combineBroad	2225.833
6	combine60ydShuttle + combineBroad	2229.043
7	combine60ydShuttle	2279.859
8	combineBench	7493.759
9	combineBroad	8988.906

```
10          combineVert 9079.765
11          combine40yd 9681.980
```

My table of AIC values by model makes it easy to compare which model performed the best. From this, you can see the two main term model, drafted ~ combine60ydShuttle + combineBench, performed the best

## ANOVA

```
anova(model2, model2_2)
```

Analysis of Deviance Table

Model 1: drafted ~ combine60ydShuttle + combineBench

Model 2: drafted ~ combine60ydShuttle \* combineBench

	Resid.	Df	Resid.	Dev	Df	Deviance	Pr(>Chi)
1	1223		1545.3				
2	1222		1544.0	1	1.3511	0.2451	

Using anova as a second method of comparison, you can see that there is not evidence at the 5% significance level to suggest a significant difference between my top two models. Therefore, I would go with model 1, drafted ~ combine60ydShuttle + combineBench because the complexity of adding interaction terms is not worth the very minimal impact.

### **drafted ~ combine60ydShuttle + combineBench**

## Practical Implications

Based on the results of my logistic regression, if I were a coach looking at this data, I would track closely my athletes' bench and 60 yd shuttle in order to see who on my team is likely to get drafted as these measurements are the best predictors.

Based on the results of my mixed effects model, I would make sure to tailor weight gain programs before the draft as you can see that different positions' linear regressions have different "baselines" for weight.

Although these models showed some significant results regarding p and z values, it is important to note that not all players that get drafted perform all/any tasks at the combine, which means my data set has a lot of NAs. Furthermore, there are also many other factors such as skill, performance history and team preferences that the models do not consider and will also significantly impact draft prospects.