In [314…
```
# import the necessary packages
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
from plotnine import *
import statsmodels.api as sm

from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, r2_score

%matplotlib inline
```

1. (5 pts) Using the heavy lifting data set from `"https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2019/2019-10-08/ipf_lifts.csv"` , build an sklearn regression model that predicts the body weight (in kg) of a compteitor. Don't standardize your variables.
   - interpret the coefficients from your model
   - describe how accurate your model is, and which metrics you used to decide that.
   - plot the residuals vs *predicted* body weight. Is the error homoskedastic? If it wasn't what issues could that cause?

</br> </br>

1. (2 pts) Build the same model as above, but z-score (`StandardScaler()`) your variables.
   - how does the interpretation of the coefficients change?

</br> </br>

1. (3 pts) Choose one of the continuous predictor variables you chose. Plot body weight by that variable using plotnine/ggplot. Is the relationship roughly linear? What do you think the consequences could be if it wasn't/isn't?

1

In [315…
```
hw = pd.read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2019/2019-10-08/ipf_lifts.csv")

hw.head()
```

Out[315…

	name	sex	event	equipment	age	age_class	division	bodyweight_kg	weight_class_kg	best3squat_kg	best3bench_kg	best3deadlift_kg
0	Hiroyuki Isagawa	M	SBD	Single-ply	NaN	NaN	NaN	67.5	67.5	205.0	140.0	225.(
1	David Mannering	M	SBD	Single-ply	24.0	24-34	NaN	67.5	67.5	225.0	132.5	235.(
2	Eddy Pengelly	M	SBD	Single-ply	35.5	35-39	NaN	67.5	67.5	245.0	157.5	270.(
3	Nanda Talambanua	M	SBD	Single-ply	19.5	20-23	NaN	67.5	67.5	195.0	110.0	240.(
4	Göran Henrysson	M	SBD	Single-ply	NaN	NaN	NaN	67.5	67.5	240.0	140.0	215.(

In [316…
```
hw.isnull().sum(axis = 0)
```

Out[316…
```
name                   0
sex                    0
event                  0
equipment              0
age                 2906
age_class           2884
division             627
bodyweight_kg        187
weight_class_kg        1
best3squat_kg      13698
best3bench_kg       2462
best3deadlift_kg   14028
place                  0
date                   0
federation             0
meet_name              0
dtype: int64
```

In [317…
```
hw = hw.dropna()  #drop missing values
hw.head()
```

Out[317…

	name	sex	event	equipment	age	age_class	division	bodyweight_kg	weight_class_kg	best3squat_kg	best3bench_kg	best3deadlift_k
208	Anna-Liisa Prinkkala	F	SBD	Single-ply	33.5	24-34	Open	44.0	44	135.0	60.0	145
209	Vuokko Viitasaari	F	SBD	Single-ply	34.5	24-34	Open	44.0	44	120.0	62.5	145
210	Maria DelCastillo	F	SBD	Single-ply	23.5	24-34	Open	44.0	44	130.0	62.5	120
211	Helen Wolsey	F	SBD	Single-ply	27.5	24-34	Open	44.0	44	112.5	60.0	135
212	Lijnie van der Holst	F	SBD	Single-ply	37.5	35-39	Open	44.0	44	105.0	65.0	130

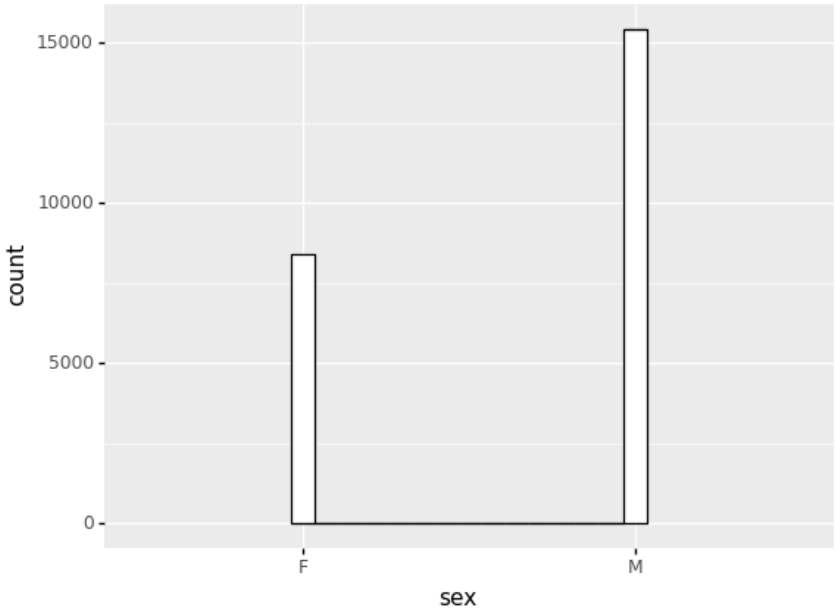
In [318…
```
hw.isnull().sum(axis = 0)  #checked to make sure is was empty
```

Out[318…
```
name              0
sex               0
event             0
equipment         0
age               0
age_class         0
division          0
bodyweight_kg     0
weight_class_kg   0
best3squat_kg     0
best3bench_kg     0
best3deadlift_kg  0
place             0
date              0
federation        0
meet_name         0
dtype: int64
```

Explore the Data

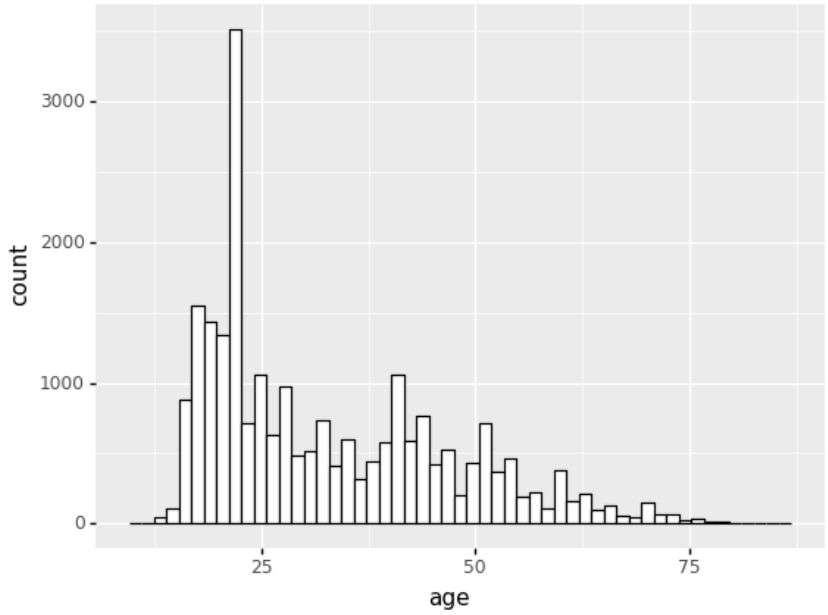
In [319…
```
(ggplot(hw, aes("sex")) + geom_histogram(fill = "white", color = "black"))


#don't use this
```



Out[319…
```
<ggplot: (-2114556327)>
```

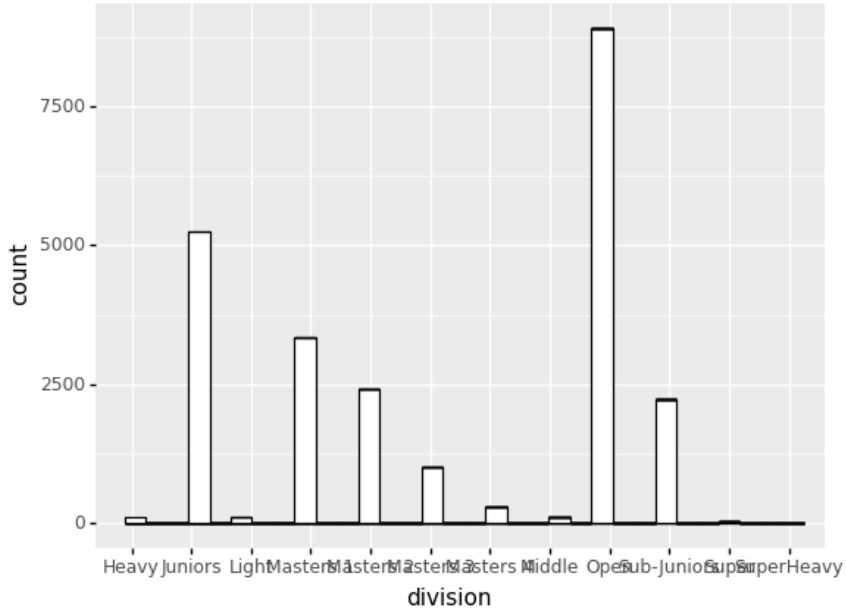
In [320…
```
(ggplot(hw, aes("age")) + geom_histogram(fill = "white", color = "black"))
```



Out[320… <ggplot: (33232406)>

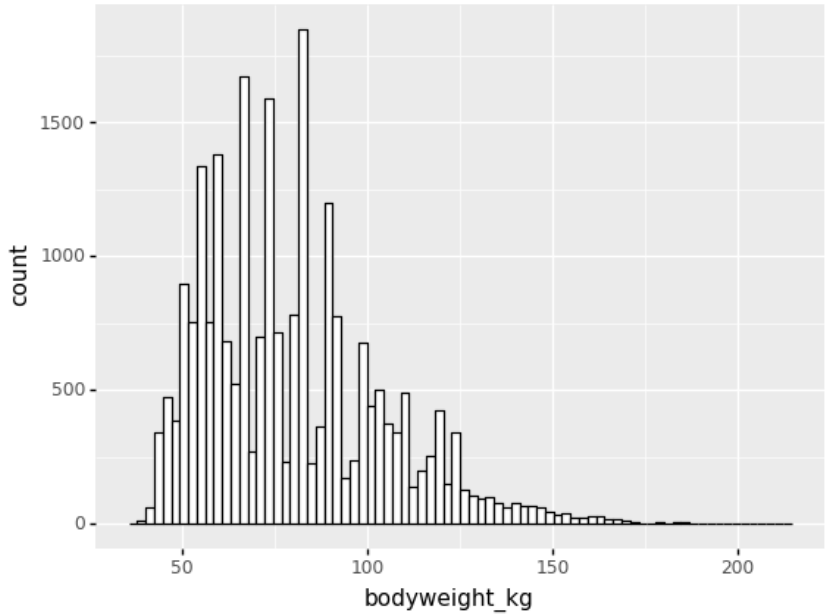
In [321…
```
(ggplot(hw, aes("division")) + geom_histogram(fill = "white", color = "black"))

#don't use this
```



Out[321… <ggplot: (-2114588282)>

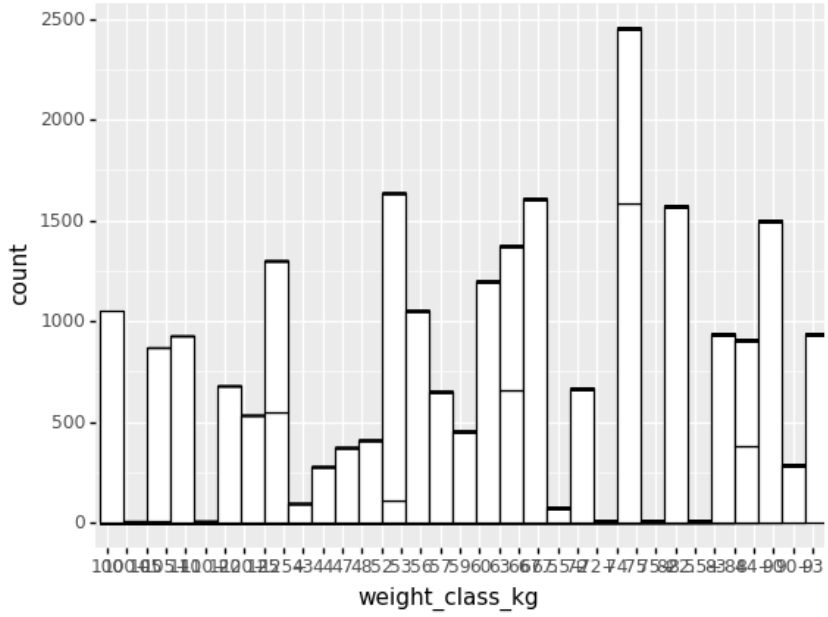
In [322…
```
(ggplot(hw, aes("bodyweight_kg")) + geom_histogram(fill = "white", color = "black"))
```



Out[322… <ggplot: (-2117448471)>

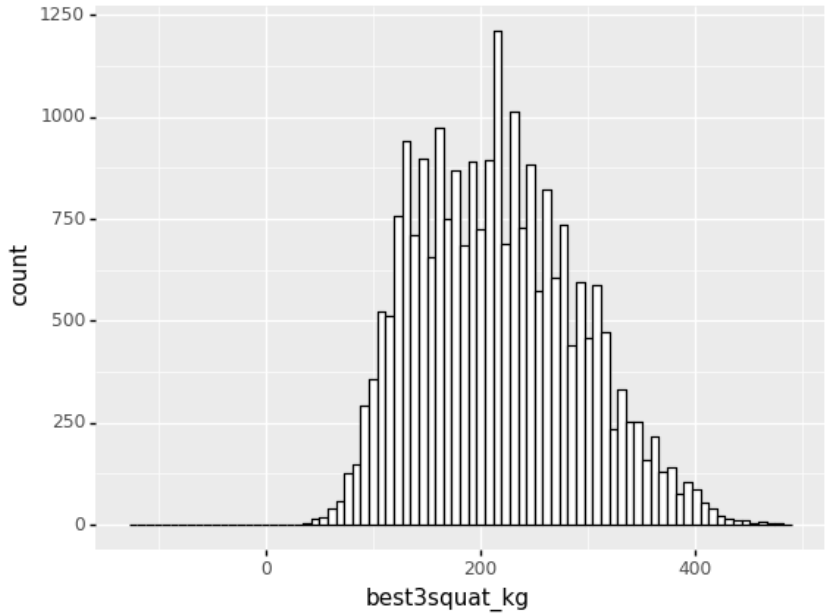
In [323…
```
(ggplot(hw, aes("weight_class_kg")) + geom_histogram(fill = "white", color = "black"))

#don't use this
```



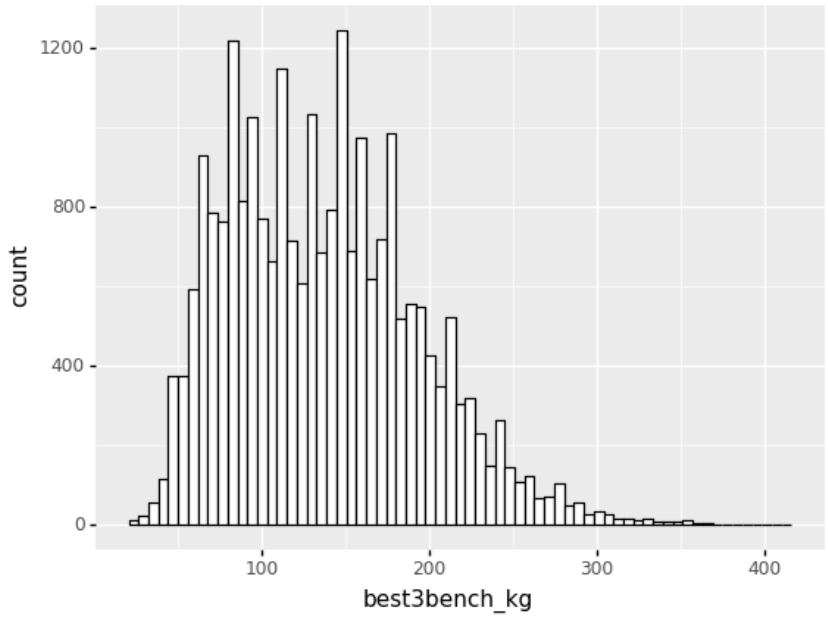
Out[323… <ggplot: (-2111896325)>

In [324…
```
(ggplot(hw, aes("best3squat_kg")) + geom_histogram(fill = "white", color = "black"))
```



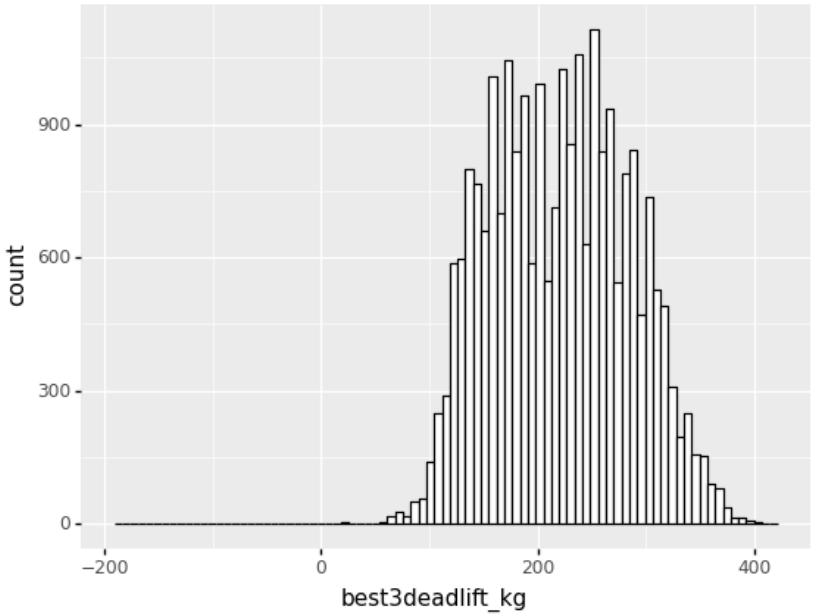
Out[324… <ggplot: (-2114613548)>

In [325…
```
(ggplot(hw, aes("best3bench_kg")) + geom_histogram(fill = "white", color = "black"))
```



Out[325… <ggplot: (-2107980241)>

In [326…
```
(ggplot(hw, aes("best3deadlift_kg")) + geom_histogram(fill = "white", color = "black"))
```



Out[326… <ggplot: (41690173)>

Model

In [327…
```
predictors = ["age", "best3squat_kg", "best3bench_kg", "best3deadlift_kg"]


X = hw[predictors]
Y = hw["bodyweight_kg"]
```

In [328…
```
LR_Model = LinearRegression()
LR_Model.fit(X,Y)
```

Out[328… LinearRegression()

Model Evaluation

In [329…
```
bodyweight_pred = hw_Model.predict(X)

mean_squared_error(Y, bodyweight_pred)
```

Out[329… 268.79218699186816

In [330…
```
bodyweight_pred[1:10]
```

Out[330…
```
array([56.28676916, 54.01476931, 53.02435635, 55.24527784, 49.84758715,
       57.83120106, 57.47769506, 57.4263512 , 55.91979119])
```

In [331…
```
r2_score(Y, bodyweight_pred)
```

Out[331… 0.548671134229698

In [332…
```
coefficients = pd.DataFrame({"Coef":LR_Model.coef_,
                "Name": predictors})
coefficients = coefficients.append({"Coef": LR_Model.intercept_,
                    "Name": "intercept"}, ignore_index = True)
```

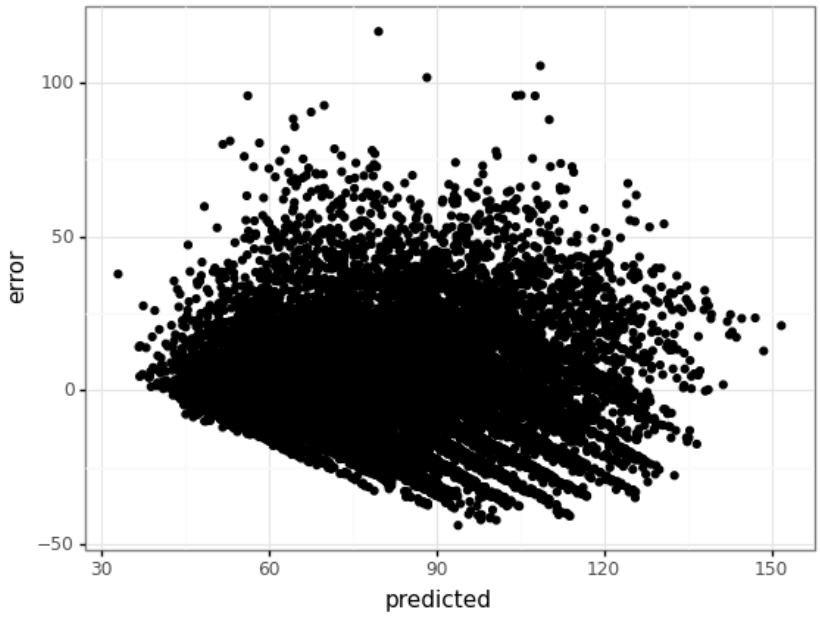
In [333…
```
coefficients
```

Out[333…

	Coef	Name
0	0.251091	age
1	0.116601	best3squat_kg
2	0.143946	best3bench_kg
3	0.027041	best3deadlift_kg
4	20.714515	intercept

In [334…
```
assump = pd.DataFrame({"error": Y - bodyweight_pred, "predicted": bodyweight_pred})

ggplot(assump, aes(x = "predicted",
                   y = "error")) + geom_point() + theme_bw()
```



Out[334… <ggplot: (41698036)>

The error seems like it doesn't have homoskedastic qualities. There are quite a few more errors near 0 than compared to the whole graph.If the graph was not homoskedastic, the model would fit a nonlinear regression and would show that this model does not fit for this data.

All of the coefficients from the model are positive values. The intercept is positive with a value of 20.714515. Of the predictors, the predictor with the greatest positive coefficient is age with a 0.251091. The smallest coefficient is best3deadlift, as it has a positive coefficient of 0.027041. With the p-values of the coeffficients for age, best3squat, best3bench, and best3deadlift all being positive yet slightly greater than zero, the regression is slightly affected positively when either of them increase. When these variables increase, the predicted outcome increases.

In terms of how accurate the model is by looking at the R-squared, the model is sort of accurate in predicting the body weight. The R-squared value is 0.548671134229698, which when mainpulating, shows us that there is about 55% variance for the model.

2

In [335…
```
#2
hw2 = pd.read_csv("https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2019/2019-10-08/ipf_lifts.csv")

hw2.head()
```

Out[335…

	name	sex	event	equipment	age	age_class	division	bodyweight_kg	weight_class_kg	best3squat_kg	best3bench_kg	best3deadlift_kg
0	Hiroyuki Isagawa	M	SBD	Single-ply	NaN	NaN	NaN	67.5	67.5	205.0	140.0	225.(
1	David Mannering	M	SBD	Single-ply	24.0	24-34	NaN	67.5	67.5	225.0	132.5	235.(
2	Eddy Pengelly	M	SBD	Single-ply	35.5	35-39	NaN	67.5	67.5	245.0	157.5	270.(
3	Nanda Talambanua	M	SBD	Single-ply	19.5	20-23	NaN	67.5	67.5	195.0	110.0	240.(
4	Göran Henrysson	M	SBD	Single-ply	NaN	NaN	NaN	67.5	67.5	240.0	140.0	215.(

In [336…
```
hw2.isnull().sum(axis = 0)
```

Out[336…
```
name                0
sex                 0
event               0
equipment           0
age              2906
age_class        2884
```

```
division              627
bodyweight_kg         187
weight_class_kg         1
best3squat_kg       13698
best3bench_kg        2462
best3deadlift_kg    14028
place                   0
date                    0
federation              0
meet_name               0
dtype: int64
```

In [337…
```
hw2 = hw.dropna()  #drop missing values
hw2.head()
```

Out[337…

	name	sex	event	equipment	age	age_class	division	bodyweight_kg	weight_class_kg	best3squat_kg	best3bench_kg	best3deadlift_k
208	Anna-Liisa Prinkkala	F	SBD	Single-ply	33.5	24-34	Open	44.0	44	135.0	60.0	145
209	Vuokko Viitasaari	F	SBD	Single-ply	34.5	24-34	Open	44.0	44	120.0	62.5	145
210	Maria DelCastillo	F	SBD	Single-ply	23.5	24-34	Open	44.0	44	130.0	62.5	120
211	Helen Wolsey	F	SBD	Single-ply	27.5	24-34	Open	44.0	44	112.5	60.0	135
212	Lijnie van der Holst	F	SBD	Single-ply	37.5	35-39	Open	44.0	44	105.0	65.0	130

Model

In [338…
```
predictors2 = ["age", "best3squat_kg", "best3bench_kg", "best3deadlift_kg"]


X = hw2[predictors2]
Y = hw2["bodyweight_kg"]
```

In [339…
```
zScore = StandardScaler()
zScore.fit(X)
Xz = zScore.transform(X)
```

In [340…
```
LR2_Model = LinearRegression()
```

In [341…
```
LR2_Model.fit(Xz,Y)
```

Out[341…
```
LinearRegression()
```

Model Evaluation

In [342…
```
bodyweight2_pred = LR2_Model.predict(Xz)

mean_squared_error(Y, bodyweight2_pred)
```

Out[342…
```
268.7921869918682
```

In [343…
```
r2_score(Y, bodyweight2_pred)
```

Out[343…
```
0.5486711342296979
```

In [344…
```
coefficients2 = pd.DataFrame({"Coef":LR2_Model.coef_,
               "Name": predictors2})
coefficients2 = coefficients2.append({"Coef": LR_Model.intercept_,
                    "Name": "intercept"}, ignore_index = True)
```

In [345…
```
coefficients2
```

Out[345…

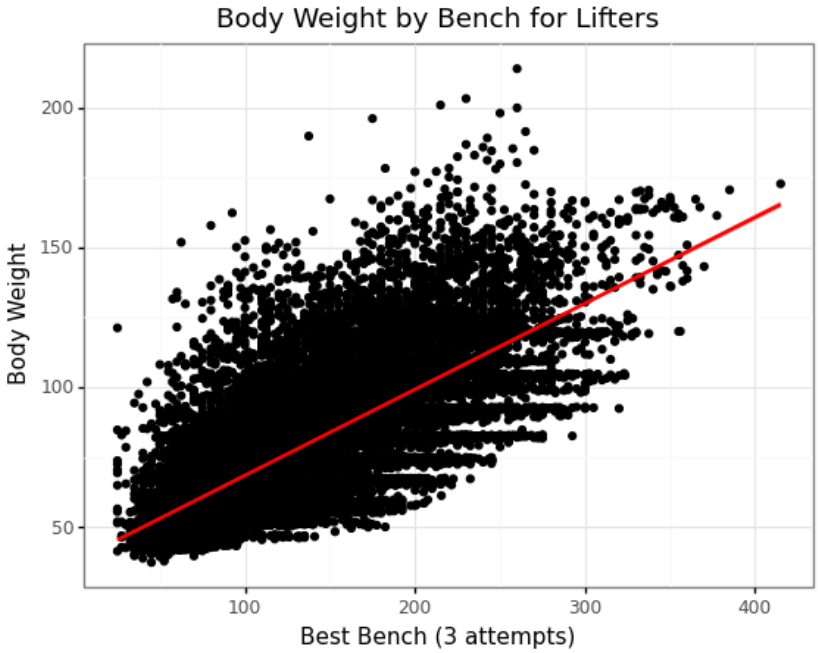
	Coef	Name
0	3.551863	age

	Coef	Name
1	8.714326	best3squat_kg
2	8.131402	best3bench_kg
3	1.723892	best3deadlift_kg
4	20.714515	intercept

When I standarized my data, my p-values increased to values that show more influence on the linear regression. Before, the values for all coefficients except the intercept were less than 1, but now they have values that are positive and greater than one. When compared to the earlier model, these coefficients have more influence on the model when increasing the variables. Now it is visible to see that age, best3squat, best3bench, and best3deadlift do have high positive p-values that influence the data based on the coefficients, but also that squat and bench data was more significant in positively affecting the data when increased than age or best deadlift. Of all the coefficients, the bestdeadlift had the lowest value of 1.723892, while bestsquat showed the highest value at 8.714326. As well, all of the coefficients have a positive value, meaning that they positively increase the weight with increases in age, bestsquat, bestbench, and bestdeadlift.

3

In [346…
```
(ggplot(hw2, aes(x = "best3bench_kg",
                 y = "bodyweight_kg"))+
                 geom_point() +
                 theme_bw() +
                 xlab('Best Bench (3 attempts)') +
                 ylab('Body Weight') +
                 ggtitle("Body Weight by Bench for Lifters") +
                 stat_smooth(fill = "red", colour = "red"))
```



Out[346… <ggplot: (39018617)>

The relationship is roughly linear for the data. If the data wasn't linear, there wouldn't be a linear regression and the data would fit non-linear regression models, which have more curvature to the regression line. If it was nonlinear, the R-squared value would not run and would throw an error, while also not having p-value coefficients.