Solutions 3 to Worked Problems

David Gerbing
The School of Business
Portland State University
gerbing@pdx.edu

- 1 Data Summary
- 2 Regression Analysis

```
from datetime import datetime as dt
now = dt.now()
print ("Analysis on", now.strftime("%Y-%m-%d"), "at", now.strftime("%H:%M"))
```

```
Analysis on 2021-07-09 at 23:45
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Data Summary

The data are the body measurements of customers for a motorcycle online clothing retailer.

Data: http://web.pdx.edu/~gerbing/data/BodyMeas.csv

```
d = pd.read_csv("http://web.pdx.edu/~gerbing/data/BodyMeas.csv")
```

a. How many samples (rows of data) and columns are there in the data file?

```
d.shape
```

```
(340, 8)
```

b. Display the first 6 rows of data and the variable names.

```
d.head()
```

	Gender	Weight	Height	Waist	Hips	Chest	Hand	Shoe
0	F	200	71	43	46	45	8.5	7.5
1	F	155	66	31	43	37	8.0	8.0
2	F	145	64	35	40	40	7.5	7.5
3	F	140	66	31	40	36	8.0	9.0
4	M	230	76	40	43	44	9.0	12.0

c. What are the variables in the data table? (From code or cut and paste.)

```
d.columns
```

```
Index(['Gender', 'Weight', 'Height', 'Waist', 'Hips', 'Chest', 'Hand', 'Shoe'], dtype='object')
```

d. Generate a frequency distribution table of Gender.

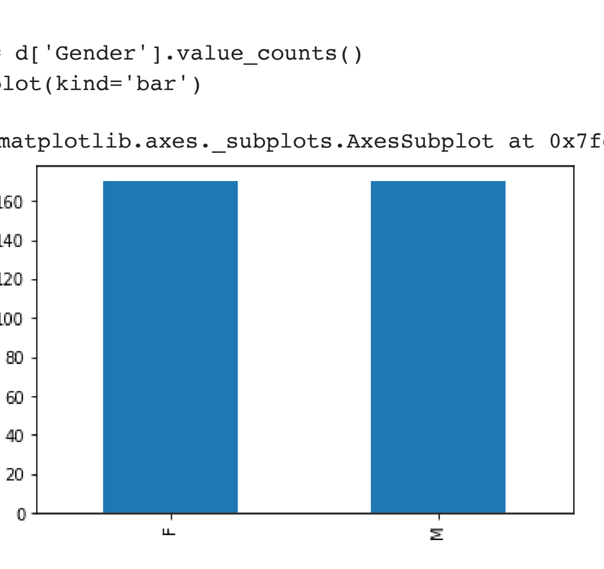
```
d['Gender'].value_counts()
```

```
F    170
M    170
Name: Gender, dtype: int64
```

e. Generate a bar chart of Gender.

```
freq = d['Gender'].value_counts()
freq.plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fecd4a8c0d0>
```



f. As with the posted online material, for this week we analyze only the Male customers for simplicity. Filter the data table to contain only customers with a Male body type. Display the first 6 rows of data and the number of rows.

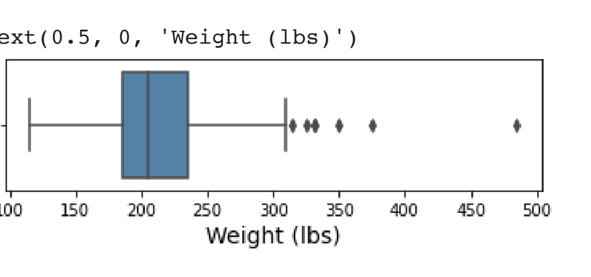
```
print(d.shape)
d = d[d['Gender'] == 'M']
print(d.shape)
```

```
(340, 8)
(170, 8)
```

g. Generate a boxplot of Weight.

```
plt.figure(figsize=(6,1.5))
sns.boxplot(x=d['Weight'], color='steelblue')
plt.xlabel('Weight (lbs)', fontsize=14)
```

```
Text(0.5, 0, 'Weight (lbs)')
```



h. There is an outlier in the data. Either the data value is mis-entered, or, if correct, any generalizations would not properly apply to this person. Subset the data frame to remove that row of data and display the number of rows in the filtered data frame to demonstrate the deletion.

There is no correct answer as to the exact number of outliers to remove. The more outliers removed, the less generalizable are the results. Particiularly if the outliers are sampled from a different distribution than the remaining data values, then they should be removed. With outliers removed, the remaining data values are typically fit better. Here remove the just the most extreme outliers, so only include reported weights below 450. If the 450lb outlier is removed, then any subsequent machine learning models will fit better, though not generalizable to people who weigh over 450lbs.

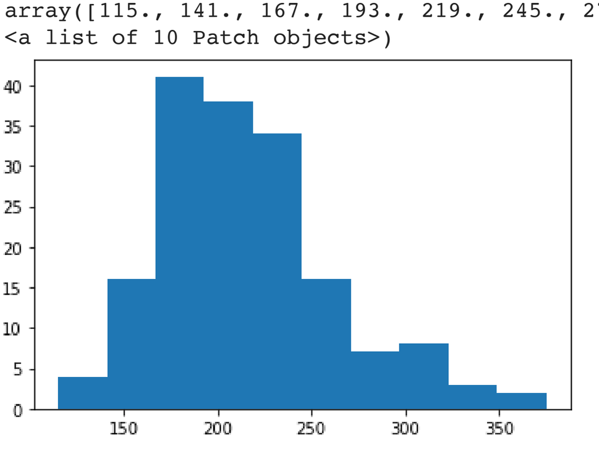
```
print(d.shape)
d = d[d['Weight'] < 400]
print(d.shape)
```

```
(170, 8)
(169, 8)
```

i. Generate a histogram of Weight.

```
plt.hist(d['Weight'])
```

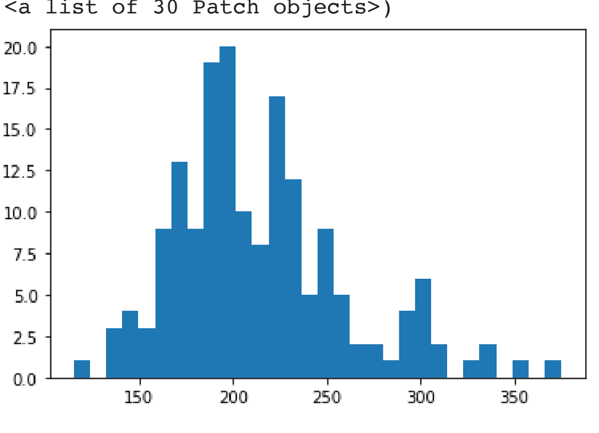
```
(array([ 4., 16., 41., 38., 34., 16.,  7.,  8.,  3.,  2.]),
 array([115., 141., 167., 193., 219., 245., 271., 297., 323., 349., 375.]),
 <a list of 10 Patch objects>)
```



j. Generate a histogram of Weight with 30 bins. Compare to the default histogram. Which is preferred?

```
plt.hist(d['Weight'], bins=30)
```

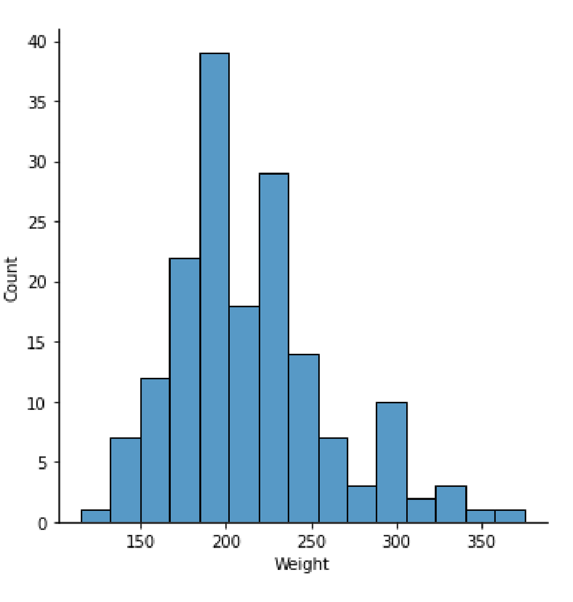
```
(array([ 1.,  0.,  3.,  4.,  3.,  9., 13.,  9., 19., 20., 10.,  8., 17.,
        12.,  5.,  9.,  5.,  2.,  2.,  1.,  4.,  6.,  2.,  0.,  1.,  2.,
         0.,  1.,  0.,  1.]),
 array([115.        , 123.66666667, 132.33333333, 141.        ,
        149.66666667, 158.33333333, 167.        , 175.66666667,
        184.33333333, 193.        , 201.66666667, 210.33333333,
        219.        , 227.66666667, 236.33333333, 245.        ,
        253.66666667, 262.33333333, 271.        , 279.66666667,
        288.33333333, 297.        , 305.66666667, 314.33333333,
        323.        , 331.66666667, 340.33333333, 349.        ,
        357.66666667, 366.33333333, 375.        ]),
 <a list of 30 Patch objects>)
```



The first histogram with the default of 10 bins is preferred becauae it likely more closely approximates the shape of the actual, smooth underlying distribution. The 30-bin histogram has too many up-and-down fluctuations.

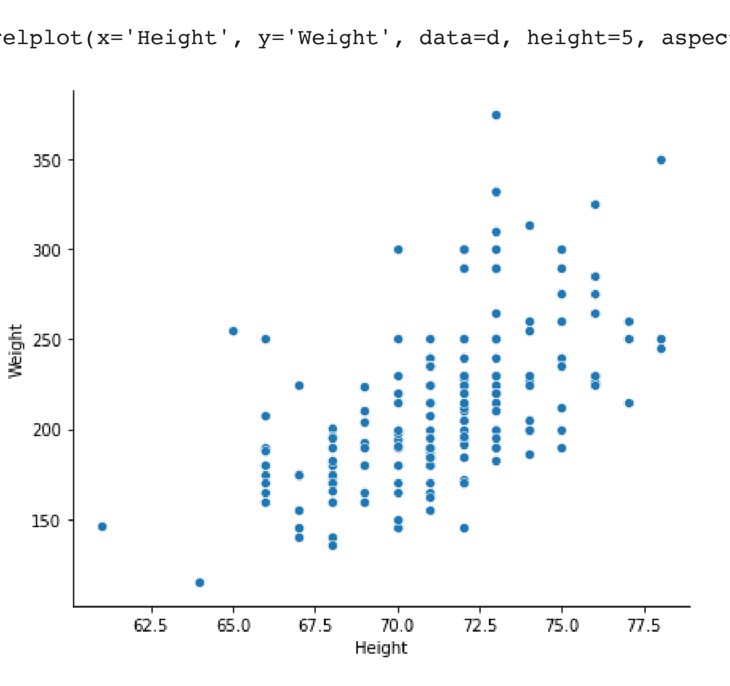
k. Generate a histogram of Weight with an overlay of a density plot.

```
sns.displot(x='Weight', data=d);
```



l. Generate the scatterplot of Weight and Height.

```
sns.relplot(x='Height', y='Weight', data=d, height=5, aspect=1.2);
```



Regression Analysis

a. Run the regression analysis of using Height to forecast Weight.

First form the X and y data structures.

```
y = d['Weight']
X = d['Chest']
```

```
import statsmodels.api as sm
from statsmodels.regression.linear_model import RegressionResults
X = sm.add_constant(X)
X.head()
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is deprec
  import pandas.util.testing as tm
```

	const	Chest
4	1.0	44
6	1.0	41
7	1.0	38
8	1.0	38
11	1.0	42

```
model = sm.OLS(y, X)
results = model.fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Weight   R-squared:                       0.796
Model:                            OLS   Adj. R-squared:                  0.795
Method:                 Least Squares   F-statistic:                     651.2
Date:                Fri, 09 Jul 2021   Prob (F-statistic):           1.63e-59
Time:                        23:45:58   Log-Likelihood:                -750.82
No. Observations:                 169   AIC:                             1506.
Df Residuals:                     167   BIC:                             1512.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       -155.2162     14.563    -10.659      0.000    -183.967    -126.466
Chest          8.2052      0.322     25.518      0.000       7.570       8.840
==============================================================================
Omnibus:                        5.370   Durbin-Watson:                   1.973
Prob(Omnibus):                  0.068   Jarque-Bera (JB):                5.088
Skew:                           0.334   Prob(JB):                       0.0785
Kurtosis:                       3.525   Cond. No.                         415.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

b. Write the regression model.

The model expresses a forecasted value of Weight, $\hat{y}_{Weight}$, in terms of a linear relationship from Chest size, x_{Chest}.

$$\hat{y}_{Weight} = -155.2162 + 8.2052(x_{Chest})$$

Note: You do not need to express with this precise notation. Improvise as needed. Such as y_wt or something.

c. For a Chest of 42 inches, what is the forecasted Weight?

```
pred = -155.2162 + 8.2052*42
print (round(pred, 1), 'lbs')
```

```
189.4 lbs
```

d. What is the residual for a person who has a chest of 42 inches and weighs 165 lbs?

resid = y - $\hat{y}$

```
resid = 165 - pred
print (round(resid, 1), 'lbs')
```

```
-24.4 lbs
```

A residual of -24.4 lb means that the person's actual *Weight* is is 24.4 lb 24.4 lb below the forecasted value.

e. Does the hypothesis test indicate a relationship between Chest and Weight?

The hypothesis test is of $\beta_1 = 0$, that is, a tesat of no population relationship between Chest and Weight with a 0 slope coefficient. The result is $t = 25.5$. That is, the sample slope coefficient of $b_1 = 8.21$ is more than 25 standard errors from the hypothesized value.

If we did repeated sampled from sample after sample we could get a distribution of b_1's about the population value. For purposes of the hypothesis test, assume that population value is 0. We would have a normal distribution of estimated (sample) slope coefficients, b_1 centered over 0. 95% of these values would be between -2 and 2 standard errors from 0 if 0 were the true value. Obtaining a value of $b_1 = 8.21$ is 25.5 standard errors from 0, a value so far from zero is an extremely improbable event, with a p-value of 0.000. Therefore, reject the null hypothesis and conclude that $beta_1 > 0$.

f. What is the best estimate of the population slope coefficient?

The 95% confidence interval for the slope coefficient is 7.57 to 8.84, the interval which likely contains the true value of the slope coefficient.

g. Interpret the confidence interval of the population slope coefficient.

With 95% confidence, for each additional inch of Chest size, on average Weight increases somewhere between 7.57 to 8.84 lbs.

h. What is the sum of the squared residuals (errors) for the best-fitting line?

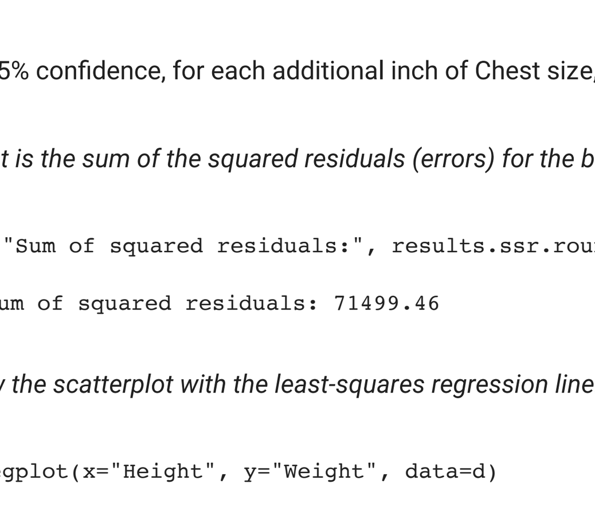
```
print("Sum of squared residuals:", results.ssr.round(2))
```

```
Sum of squared residuals: 71499.46
```

i. Show the scatterplot with the least-squares regression line.

```
sns.regplot(x="Height", y="Weight", data=d)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fecd3905290>
```



j. What is the standard deviation of the residuals. Interpret.

For a normal distribution, 95% of the values are with two standard errors of its mean, a range of four standard deviations.

```
RMSE = np.sqrt(results.mse_resid)
print("Stdev of residuals:", RMSE.round(2))
res_range = 4 * RMSE
print("95% range of residuals:", res_range.round(2))
```

```
Stdev of residuals: 20.69
95% range of residuals: 82.77
```

A range of 152 lbs for 95% of the data values about their respective points on the regression line indicates a fair amount of scatter. Presumably a better model could be constructed, perhaps by adding more predictor variables to the model.

Plus, this scatter describes the variability of the training data for the model that is optimized on this data. Going to new data to forecast means even more scatter as the model is not optimized on that new data.

k. What is R-squared? Interpret.

```
print("R-squared:", results.rsquared.round(3))
```

```
R-squared: 0.796
```

Not terrible, as $R^2 = 0.31$ indicates better fit, a better understanding of Weight given Chest size, than the null model without Chest size. That said, a better fit is desired. Usually adding more relevant predictor variables to the model can increase fit.