STA302: Assignment 3

Craig Burkett

Due: Dec 3, 2015

Due at the beginning of lecture on Thursday, Dec 3rd. Please hand it in on 8.5 x 11 inch paper, stapled in the upper left, with no other packaging and no title page. Please try to make this assignment look like something you might hand in to your boss at a job. In particular, it is inappropriate to hand in pages of R output without explanation or interpretation. Do not print out your datasets. Quote relevant numbers from your R output as part of your solutions but do not output your code in the body of your report. The only direct R output you should submit with the assignment are relevant plots. ***You must append your R program file to the end of the assignment, formatted nicely with a fixed-width font.*** *No assignment will be marked without a program file, and marks will be deducted if the instructions above are not followed.*

In this assignment, you will investigate a dataset involving 100m performances for Male and Female track and field athletes. You will also explore the US States dataset mentioned in lecture. Any time that I use the words {Present, State, Give, Show, Predict, Display}, you must supply that graphic in your submission. If I say {Produce, Make}, you do not need to show what you produced or made, but you still need to do it. Benchmarks of 95% confidence and 5% significance can be used unless otherwise specified. The US State data dictionary is available here.

A US States (25 marks)

Load the US State data the same way I did in my sample code. You'll have to bind the regions to the states data frame; you can find them in the object `state.division`. There should be 9 levels for this factor.

1. It's usually a good idea to look at some summary statistics before building a model.

   (a) Compute a new column called $Density = 1000 * \frac{Population}{Area}$.
   Direct enough to do with the one line command
   `states$Density = states$Population * 1000 / states$Area`

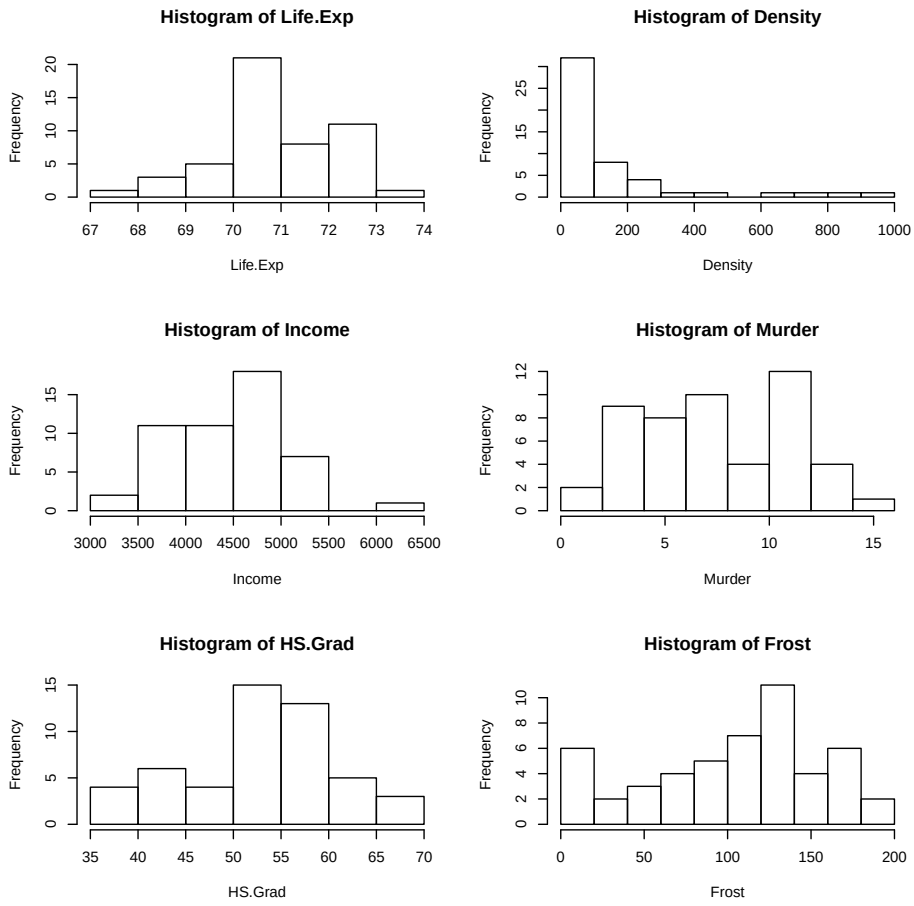
   (b) Present a correlation matrix showing the correlations between Life Expectancy, Density, Income, Murder, HS Grad and Frost, in that order. (2 mks)
   The correlation matrix is in Table 1.

Table 1

	Life.Exp	Density	Income	Murder	HS.Grad	Frost
Life.Exp	1.00	0.09	0.34	-0.78	0.58	0.26
Density	0.09	1.00	0.33	-0.18	-0.09	0.00
Income	0.34	0.33	1.00	-0.23	0.62	0.23
Murder	-0.78	-0.18	-0.23	1.00	-0.49	-0.54
HS.Grad	0.58	-0.09	0.62	-0.49	1.00	0.37
Frost	0.26	0.00	0.23	-0.54	0.37	1.00

   (c) We might need to do some transformations down the road. In preparation, present a histogram of each of these six numerical variables, laid out in a 3x2 grid. (2 mks)
   These histograms are in Figure 1.

Figure 1



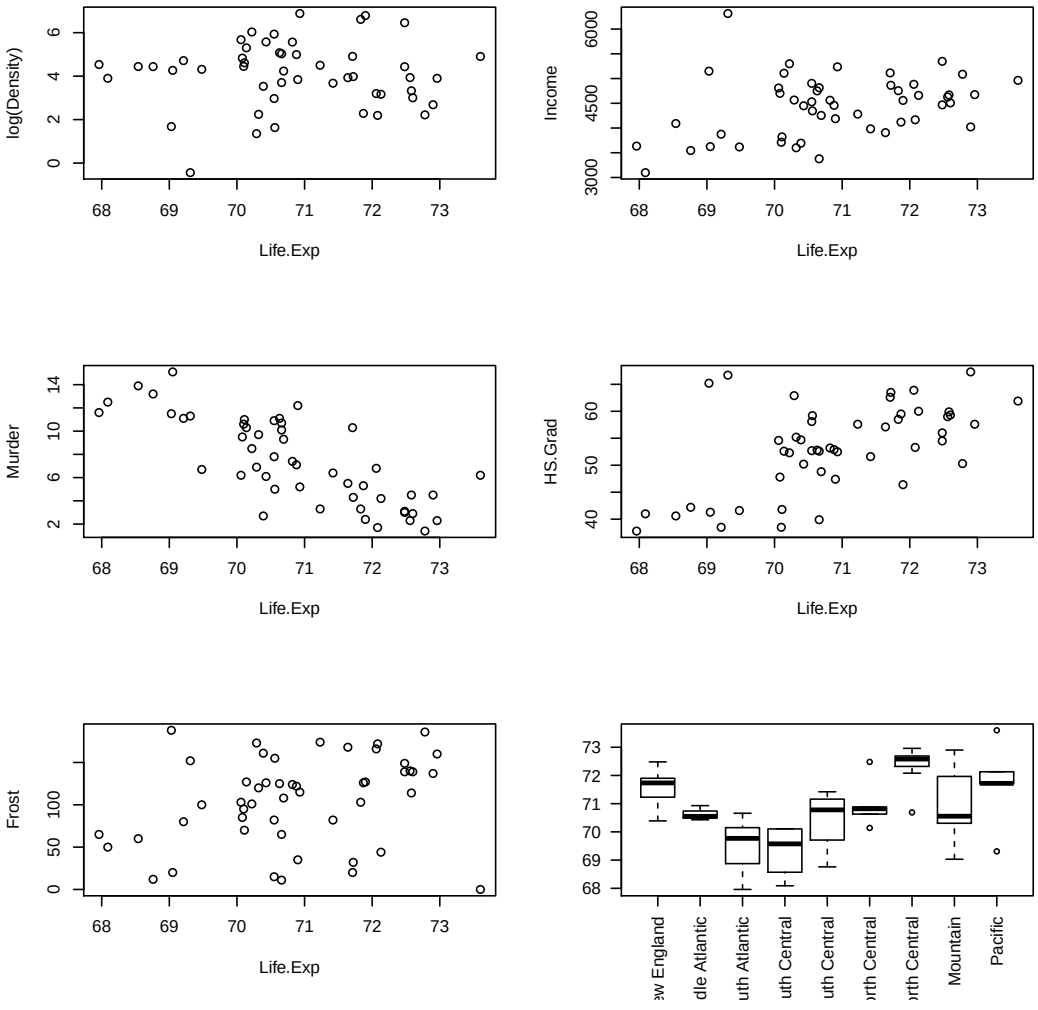
(d) It's nice to know what the bivariate relationships look like. Produce a scatterplot of Life Expectancy vs. each of Density, Income, Murder, HS Grad and Frost. For a sixth plot, make a boxplot with Life Expectancy along the y-axis and Region along the x-axis. You can use `with(df, boxplot(y ~ x))` to accomplish this, and you do not need to show these 6 plots in your submission, because ...

You didn't need to show this!

(e) It looks like we're in need of some transformations. Redo the last six plots, this time using the log of Density. Present your plots in a 3x2 layout, on one full page in your report. (4 mks)

These scatterplots are in Figure 2.

Figure 2


log(Density)
Life.Exp
Income
Life.Exp
Murder
Life.Exp
HS.Grad
Life.Exp
Frost
Life.Exp
Mountain
Pacific

(f) Everything looks good, we just need to make some indicator variables for the factor (Region). Actually, R can do this for us, but it's a good exercise. Make indicators (1s and 0s) for all of the nine levels. You can store them as columns in the data frame.

Included in the code appendix. The relevant portion is

```
    states <- within(states, {
  I_esc<- ifelse(Region == Regs[1], 1, 0)
  I_p  <- ifelse(Region == Regs[2], 1, 0)
  I_m  <- ifelse(Region == Regs[3], 1, 0)
  I_wsc<- ifelse(Region == Regs[4], 1, 0)
  I_ne <- ifelse(Region == Regs[5], 1, 0)
  I_sa <- ifelse(Region == Regs[6], 1, 0)
  I_enc<- ifelse(Region == Regs[7], 1, 0)
  I_wnc<- ifelse(Region == Regs[8], 1, 0)
  I_ma <- ifelse(Region == Regs[9], 1, 0)
})
```

2. Now it's time to fit an MLR model. Let's try to predict the Life Expectancy of a State using other demographic information.

   (a) Fit a linear model with Life Expectancy as the response, and log(Density), Income, Murder, HS graduation rate, Frost, and all of the Region indicators as predictors.

   The model summary is in Table 2. It is naive to include all the indicators since they are linearly dependent (the sum in any row is 1), which is why the last one cannot be estimated.

   (b) Oops, that was naive of us. If we want to use all of these indicators, we should fit a model with no intercept. Do that, and present the coefficient table (from summary.lm) using a fixed-width font. (1 mk)

   Those p-values (for the factor) aren't too helpful with a no-intercept model. But, looking at the boxplots, it seems like we could get away with having an indicator solely for West North Central. Fit a reduced model with only that indicator (and all the other predictors too), and an intercept. The inclusion of an intercept will group the other eight regions into one group.

   The model summary is in Table 3.

   The reduced model with only West North Central is in Table 4.

Table 2

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	68.6757	1.4564	47.15	0.0000
log(Density)	0.3271	0.1406	2.33	0.0258
Income	0.0001	0.0003	0.35	0.7308
Murder	-0.2310	0.0457	-5.05	0.0000
HS.Grad	0.0472	0.0292	1.62	0.1145
Frost	-0.0029	0.0040	-0.72	0.4783
I_ma	-0.9406	0.6559	-1.43	0.1602
I_wnc	0.7430	0.6359	1.17	0.2503
I_enc	-0.1393	0.5890	-0.24	0.8143
I_sa	-0.8711	0.4538	-1.92	0.0629
I_ne	-0.6597	0.6570	-1.00	0.3220
I_wsc	0.5362	0.5007	1.07	0.2913
I_m	0.2569	0.6805	0.38	0.7080
I_p	0.2856	0.7936	0.36	0.7210
I_esc	NA	NA	NA	NA

(c) Perform a partial F-test to see if the eight other regions are equivalent with respect to mean life expectancy. Give the test statistic, df, and p-value, as well as a conclusion in plain English. (4 mks)

The partial F-test is given in Table 5. The second row corresponds to the test of interest.

Recall that the null hypothesis for this test is that the two models are equivalent, ie: the eight other regions are equivalent with respect to mean life expectancy. The p-value is 0.1141, suggesting that there is no evidence against this hypothesis. Based on this, we can conclude that the eight other regions are equivalent with respect to mean life expectancy.

Statistically, this means we can use the reduced model.

```
Analysis of Variance Table
```

Model 1: Life.Exp log(Density) + Income + Murder + HS.Grad + Frost + I_wnc

Model 2: Life.Exp log(Density) + Income + Murder + HS.Grad + Frost + I_ma + I_wnc + I_enc + I_sa + I_ne + I_wsc + I_m + I_p + I_esc - 1

(d) Good, let's go with this reduced model. Identify the non-significant predictors (at the 5% level) and fit another reduced model with-

Table 3

	Estimate	Std. Error	t value	Pr(>|t|)
log(Density)	0.3271	0.1406	2.33	0.0258
Income	0.0001	0.0003	0.35	0.7308
Murder	-0.2310	0.0457	-5.05	0.0000
HS.Grad	0.0472	0.0292	1.62	0.1145
Frost	-0.0029	0.0040	-0.72	0.4783
I_ma	67.7351	1.7389	38.95	0.0000
I_wnc	69.4187	1.5667	44.31	0.0000
I_enc	68.5364	1.7135	40.00	0.0000
I_sa	67.8046	1.5214	44.57	0.0000
I_ne	68.0160	1.7047	39.90	0.0000
I_wsc	69.2119	1.4482	47.79	0.0000
I_m	68.9326	1.6968	40.63	0.0000
I_p	68.9613	1.7003	40.56	0.0000
I_esc	68.6757	1.4564	47.15	0.0000

Table 4

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	68.9443	1.3451	51.26	0.0000
log(Density)	0.1529	0.0934	1.64	0.1091
Income	-0.0001	0.0002	-0.38	0.7022
Murder	-0.2200	0.0433	-5.08	0.0000
HS.Grad	0.0727	0.0229	3.18	0.0028
Frost	-0.0063	0.0025	-2.51	0.0158
I_wnc	0.9492	0.3481	2.73	0.0092

out them. Compare to the model from the previous part with a partial F-test. Give p-value and conclusion. (3 mks)

The nonsignificant ones would be the log density and the incom e, refer back to Table 4. The nested model test is in Table 6.

Analysis of Variance Table

Model 1: Life.Exp log(Density) + Income + Murder + HS.Grad + Frost + I_wnc

Model 2: Life.Exp Murder + HS.Grad + Frost + I_wnc

Just for fun, the summary table for the new reduced model is in Table 7.

Table 5

Model	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	43	21.28				
2	36	15.73	7	5.56	1.82	0.1141

Table 6

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	45	22.71				
2	43	21.28	2	1.43	1.44	0.2476

(e) Write out the equation for your final model, including the fitted coefficients you estimated using Least Squares. It should look like this except with actual numbers and variable names: (2 mks)

$$\hat{Y} = b_0 + b_1 \cdot X_1 + ...$$

The fitted model equation is

$$\begin{aligned} \text{Expected Life Expectancy} = & 70.55 \\ & - 0.25 \times \text{Murder} \\ & + 0.05 \times \text{HS.Grad} \\ & - 0.01 \times \text{Frost} \\ & + 0.75 \times \text{I\_wnc} \end{aligned}$$

(f) Give an interpretation, in plain English, of the coefficient of Murder. (1mk)

For each murder per 100,000 persons, average life expectancy decreases by a quarter (0.25) of a year.

(g) Give an interpretation, in plain English, of the coefficient of the indicator of West North Central region. (1mk)

West North Central states have, on average, a life expectancy that is 0.75 years higher than the average for other (non West North Central) states.

3. Let's check the assumptions for this final model.

(a) Show a plot of residuals vs. fitted values and a Normal QQ plot for your MLR model, on a 1x2 layout. Do you have any major

Table 7

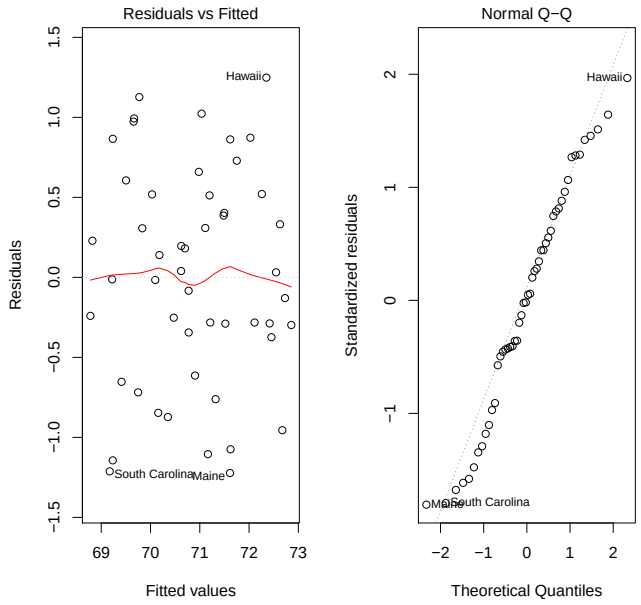
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	70.5494	0.9642	73.17	0.0000
Murder	-0.2534	0.0374	-6.77	0.0000
HS.Grad	0.0545	0.0147	3.71	0.0006
Frost	-0.0077	0.0024	-3.24	0.0022
I_wnc	0.7484	0.3259	2.30	0.0264

concerns with them? If yes, say what your concerns are. If not, say why the plots look OK. (2 mks)

The residual vs fitted values plot shows no fanning and no trend / pattern overall. The Normal Q-Q plot shows some deviations at the tails, more significant for the lower quantiles.

In Figure 3.

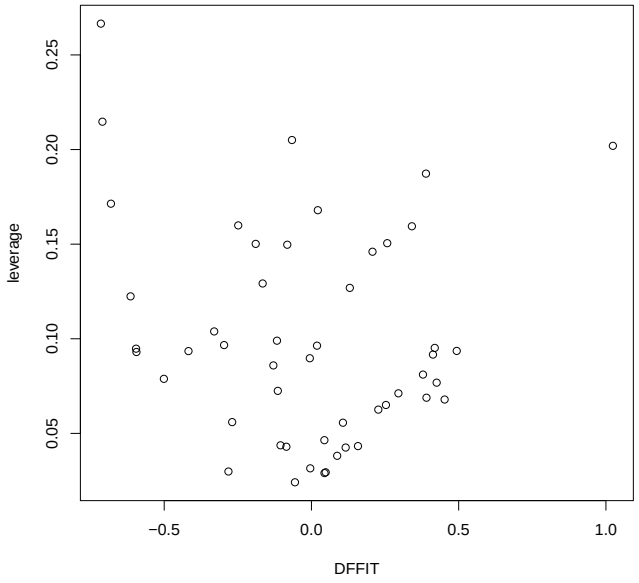
Figure 3



(b) Show a plot of the leverage vs. DFFITs (absolute value). (1 mk)

In Figure 4.

Figure 4



(c) There looks to be a point with pretty high DFFIT. Which state is that? (1 mk)

Hawaii.

(d) There looks to be a point with pretty high leverage. Which state is that? (1 mk)

Nevada.

B Athletics (20 marks)

Now let's shift our attention over to the Athletics dataset.

1. Let's prepare the 100m data for analysis.

   (a) Read in the two files, and add a column to each indicating the sex of the athlete. Bind the two files together.

   The new columns can be added as simply as this, as seen in the appendix code.

```
tnf100M$sex <- tnf1500M$sex <- "Male"
tnf100W$sex <- tnf1500W$sex <- "Female"
```

   (b) Some of the times were run at high altitude, and have an 'A' in the time. Delete all of these rows, and convert the time to numeric.

   (c) Keep only the columns country, date, birth, wind, time and sex.

   (d) You'll have to format those dates, but you'll run into a problem with the 2-digit years. R thinks '69' and later is 1900s, and '68' and earlier is 2000s. There are a few ways to fix this; any of them are fine.
   Hint: subtract 100 years from all dates in a certain range, or append the century prefix based on a condition.

   (e) Make two factors in your data frame for the Year and Month the race was run.

   (f) Once that is done, compute the age of the athlete (in Years) and store as a column. You should note that there are 365.25 days in a year, on average. Delete rows for athletes with a missing age.

   (g) We've also got to fix the wind measurements. First, replace all commas with a decimal point.

   (h) Next, readings close to zero have a ± sign beside them in this file. You could try to remove it, or just search for numbers that also have '0.0' and overwrite them with 0. That should leave you with all numbers, plus a few missing values. Delete rows without wind.

   (i) Finally, 100m times are highly dependent on wind. In order to compare them, we should adjust the times for wind. A positive wind is a tailwind, and it's estimated to add 0.05s to a man's time, for every 1.0 m/s of wind (0.06s to a woman's time). Convert the

times to what they would equivalently be with no wind (so, for example, 11.00s in a +1.5m/s wind becomes 11.09s for a female). The time could also get faster if it's a headwind (negative).

For general preparation of the data, see the code block in the appendix, beginning with the comment `# Format 100m data` .

2. Let's prepare the 1500m data for analysis.

   (a) Read in the two files, and add a column to each indicating the sex of the athlete. Bind the two files together.

   (b) Some of the times were run at high altitude, and have an 'A' in the time. Delete all of these rows, and convert the time to numeric. Keep only the columns country, date, birth, time and sex. This link might be helpful if you're having trouble converting the times (the format is different than 100m times).

   (c) Format the dates as above (with Year and Month factors too) and compute the age of the athlete (in Years) and store as a column. There seems to be one women who ran a race 3 years before she was born. That's certainly a typo, so delete that row. At this point, you can delete anyone whose age is missing as well.

   (d) You should convert the times to seconds as well. R stores time objects as a time and a date (the current date you created the time) so if you just subtract '00:00' from your time and cast as numeric, you'll have the time in minutes.

   (e) Luckily, wind is not a factor in this race. Well, it is, but it's not recorded and it rarely helps.

   For general preparation of the data, see the code block in the appendix, beginning with the comment `# Format 1500m data` .

3. Now you have two identical data frames (if you ignore wind from the 100m data). Bind these data frames together into one frame, without wind. Before you do that, you should create a column in each called Race with an appropriate label, so that you can tell which times came from which race. Present the summary statistics obtained from `summary()` for this finished data frame, neatly formatted in your report. (6 mks)

   Adding the columns, again as simple as (appendix code)

```
tnf100$race <- "100m"
tnf1500$race <- "1500m"
```

Binding and classifying the race as a factor done with (appendix code)

```
tnf <- rbind(subset(tnf100, select=-wind), tnf1500)
tnf$race <- factor(tnf$race)
```

This summary table is in Table 8-9

Table 8

	country	date	birth	time
1	USA :2773	Min. :1958-08-28	Min. :1936-06-13	Min. : 9.625
2	KEN :1920	1st Qu.:1993-05-21	1st Qu.:1966-03-21	1st Qu.: 11.042
3	JAM : 909	Median :2001-08-17	Median :1975-11-02	Median :215.280
4	GBR : 749	Mean :2000-08-08	Mean :1974-08-23	Mean :155.623
5	RUS : 726	3rd Qu.:2009-08-15	3rd Qu.:1984-04-16	3rd Qu.:217.430
6	MAR : 663	Max. :2015-09-26	Max. :1999-05-06	Max. :244.990
7	(Other):6400			

Table 9

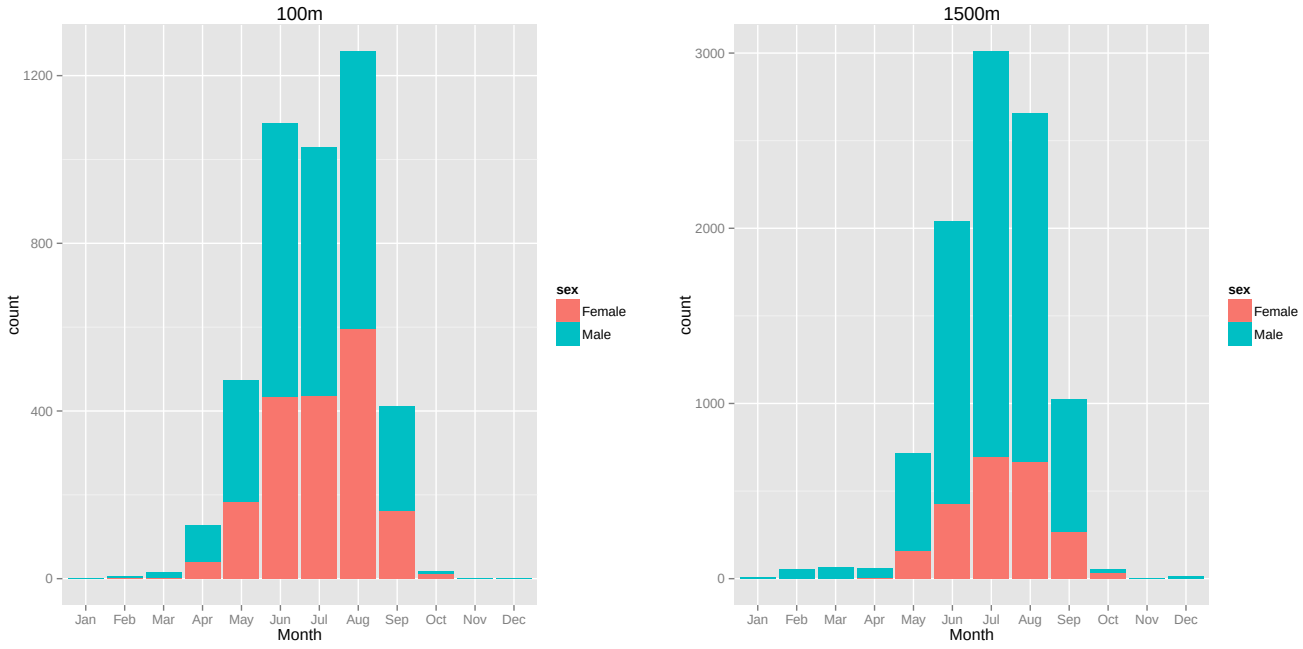
	sex	ageYrs	Month	Year	race
1	Female: 4130	Min. :15.53	Jul :4040	2012 : 674	100m :4430
2	Male :10010	1st Qu.:23.19	Aug :3914	2015 : 640	1500m:9710
3		Median :25.60	Jun :3127	2013 : 591	
4		Mean :25.96	Sep :1440	2008 : 555	
5		3rd Qu.:28.40	May :1188	2011 : 548	
6		Max. :44.23	Apr : 188	2009 : 540	
7			(Other): 243	(Other):10592	

4. Let's see if there's a season for running fast times in each of these races. Give a barplot (grouped by sex) for each race, showing the counts of top times by Month, in order, aggregated over all years. You should have two plots; one for each race. Make sure all of the months appear on the x-axis. *Hint: If you're using* `ddply()`, *you can pass it an argument* `.drop = F` *so the unused levels don't get dropped.* (3 mks)

The 100m race is in Figure 5 (a), and the 1500m in (b). Unsurprisingly, the running season is June, July, and August, and to a lesser extent includes May and September. (It's OK if these plots are superimposed instead of stacked).

Figure 5

(a) (b)



5. Let's have a look at what countries are producing fast times, for each race and sex. Present a separate two-way table for each race with Country along the rows and Sex across the columns, showing the counts of top times. Don't show any countries that don't have counts for either sex, and order the countries in decreasing order of total count for both sex. Show only the top 25 countries for each race. (3 mks)

   These are in Table 10 (100m) and 11 (1500m).

6. We will examine the relationship between 100m and 1500m performance for a country, to see if they are related.

   (a) Make a new data frame, containing a column for Country, a column for Sex, and two columns with the total counts of top times

Table 10

	race	country	countM	countW	count
1	100m	USA	1047	703	1750
2	100m	JAM	500	407	907
3	100m	TTO	177	48	225
4	100m	NGR	100	69	169
5	100m	GBR	155	5	160
6	100m	FRA	69	66	135
7	100m	BAH	17	108	125
8	100m	CAN	100	8	108
9	100m	GDR	1	81	82
10	100m	SKN	70	0	70
11	100m	UKR	2	68	70
12	100m	RUS	0	68	68
13	100m	GER	6	60	66
14	100m	NAM	57	0	57
15	100m	BUL	0	42	42
16	100m	CIV	5	29	34
17	100m	GHA	34	0	34
18	100m	BAR	31	0	31
19	100m	GRE	0	31	31
20	100m	POR	28	0	28
21	100m	CHN	11	12	23
22	100m	ANT	21	0	21
23	100m	NED	7	14	21
24	100m	QAT	19	0	19
25	100m	JPN	17	0	17

for each race. Subset this data frame by country/sex pairings that have at least one count for **both races**. If you did this correctly, you should have 45 rows.

Look at the start of the appendix code with comment

`# Regression - Q6.`

(b) Present a scatterplot showing the log of counts for 100m top times along the x-axis, and log of counts for 1500m top times along y. Use a different colour (shade) and plotting symbol for male and female counts, and put the line of best fit for each sex through the respective points. This line should be coloured the same way (or

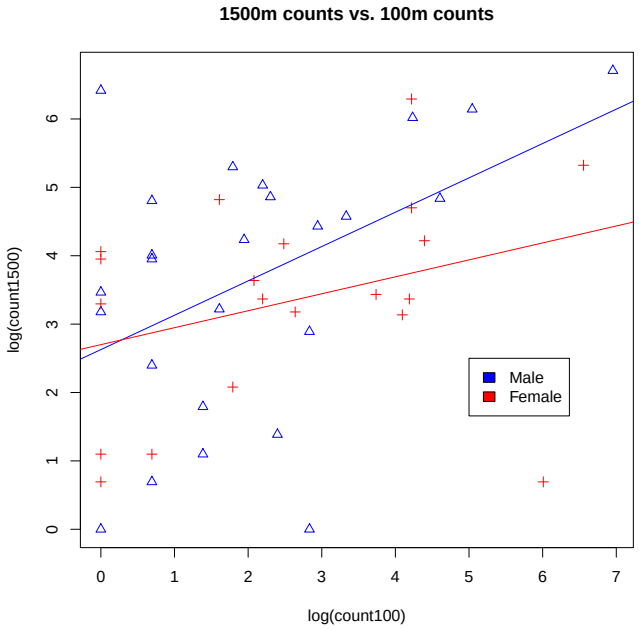
Table 11

	race	country	countM	countW	count
1	1500m	KEN	1795	125	1920
2	1500m	USA	818	205	1023
3	1500m	MAR	612	50	662
4	1500m	RUS	118	540	658
5	1500m	GBR	465	124	589
6	1500m	ESP	490	52	542
7	1500m	FRA	411	29	440
8	1500m	ALG	318	24	342
9	1500m	ETH	206	102	308
10	1500m	ROU	1	225	226
11	1500m	GER	200	23	223
12	1500m	IRL	149	29	178
13	1500m	AUS	153	21	174
14	1500m	NZL	170	1	171
15	1500m	UKR	55	110	165
16	1500m	CAN	126	38	164
17	1500m	ITA	122	20	142
18	1500m	BRN	81	54	135
19	1500m	RSA	129	3	132
20	1500m	POR	97	33	130
21	1500m	GDR	32	68	100
22	1500m	NED	69	24	93
23	1500m	SUI	68	18	86
24	1500m	QAT	84	0	84
25	1500m	FRG	74	8	82

shade or line type if you don't want to print in colour) and the plot should have a legend. (4 mks)

The scatterplot is in Figure 6.

Figure 6



(c) Finally, fit a model regressing the log of 1500m counts on the log of 100m counts and sex. Start off with a model with both main effects and an interaction, and test the interaction with a partial F-test. If it is not significant, remove it and test the additive model, removing anything that is not significant. Give the fitted equation for your final model and interpret the parameter estimate(s) in plain English. (4 mks)

The summary table for the model with interaction is in Table 12 with the anova table in Table 13. Note the equivalence, and that the interaction is not significant. Removing this (and sex, also insignificant), yields a model only with log(count100) - the total count of 100m fast time runs. The final summary table is Table 14. The fitted equation is

$$\log(\text{count1500}) = 2.71 + 0.35 \log(\text{count100})$$

Exponentiating

$$\text{count1500} = \exp(2.71) \times \text{count100}^{0.35}$$

The interpretation is that the log count of 1500m fast times increases on average by 0.35 for every increase in log count of 100m fast times.

The non-log scale interpretation relates to power-law increases, with a power-coefficient of 0.35.

Table 12

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.7000	0.5996	4.50	0.0001
log(count100)	0.2479	0.1847	1.34	0.1869
sexMale	-0.0725	0.7906	-0.09	0.9274
log(count100):sexMale	0.2544	0.2662	0.96	0.3449

Table 13

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
log(count100)	1	19.81	19.81	7.15	0.0107
sex	1	2.84	2.84	1.02	0.3175
log(count100):sex	1	2.53	2.53	0.91	0.3449
Residuals	41	113.60	2.77		

Table 14

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.7115	0.3886	6.98	0.0000
log(count100)	0.3527	0.1318	2.68	0.0105

C Format (5 marks)

Please make your submission look nice. This means:

- Proper sentences free of typing errors
- Graphs and tables should be in the **body of the report**, not thrown in at the end
- R Code should be appended to the assignment, in a small fixed-width font like `courier new`

  Missing any of these items will forfeit all of the format marks.