10.16

10.16 The Tennis Modelling Challenge

Bradley-Terry Model

Question 1

We are given

$$\mathbb{P}(\text{Player } a \text{ wins a match against Player } b) = \frac{\exp(\beta_a - \beta_b)}{1 + \exp(\beta_a - \beta_b)} \tag{1}$$

so

$$\begin{aligned}\mathbb{P}(\text{Player } b \text{ wins a match against Player } a) &= \frac{\exp(\beta_b - \beta_a)}{1 + \exp(\beta_b - \beta_a)} \\ &= \frac{1}{1 + \exp(\beta_a - \beta_b)}.\end{aligned} \tag{2}$$

Therefore this Generalised Linear Model is a Bernoulli distribution where

$$Y_i \sim \text{Bernoulli}(\mu_i) \quad , \ \mu_i = \mathbb{P}(\text{Player } a_i \text{ wins a match against Player } b_i)$$

so we have the exponential family

$$\begin{aligned}f(y_i; \mu_i) &= \exp\left[\ln\left(\frac{\mu_i}{1-\mu_i}\right) y_i + \ln(1-\mu_i)\right] \\ &= \exp[y_i \ln(\mu_i) + (1 - y_i)\ln(1-\mu_i)]\end{aligned}$$

which has canonical link function

$$g(\mu_i) = \ln\left(\frac{\mu_i}{1-\mu_i}\right) = \text{logit}(\mu_i).$$

We should quickly note that logit is an invertible, monotonic function from $(0, 1)$ to $\mathbb{R}$ with

$$\text{logit}^{-1}(\theta) = \text{logistic}(\theta) = \frac{e^\theta}{1 + e^\theta}$$

thus from the definition of our model, we have that

$$\begin{aligned}\mathbb{P}(\text{Player } a_i \text{ wins a match against Player } b_i) = \mu_i = \text{logistic}\big(x_i^T \beta\big) &= \frac{\exp\big(x_i^T \beta\big)}{1 + \exp\big(x_i^T \beta\big)} \\ &= \frac{\exp(\beta_{a_i} - \beta_{b_i})}{1 + \exp(\beta_{a_i} - \beta_{b_i})}\end{aligned}$$

Thus we deduce

$$x_i^T \beta = \beta_{a_i} - \beta_{b_i}$$

where x_i^T represents the i^{th} row of the design matrix, so the i^{th} row of the design matrix has an entry of 1 for player a_i, -1 for player b_i and 0 otherwise.

Exchanging the order of players in Equation 1 gives Equation 2, which we see is of the same form. Going through the above derivation again with a_i and b_i swapped, we see that the exponentially family and canonical link function remain the same, but all the entries in the design matrix are swapped (i.e. all entries that were previously 1 becomes -1 and vice versa).

Question 2

Fixing the coefficient for "Agassi A." at 0 is equivalent to removing the column for "Agassi A." from the design matrix. Using the data from the period 2000-2014 we get the coefficients shown in Appendix A.1.1.

Note that the choice of what coefficient we fix does not change the fitted values μ_i as the fitted values depend on the difference between distinct coefficients, so fixing the i^{th} coefficient at 0 would be equivalent to subtracting the i^{th} coefficient from all coefficients.

Now we can calculate the logistic loss using

$$\begin{aligned}
\text{Logistic Loss} &= -\frac{1}{N}\sum_{i=1}^{N} \ln f(y_i, \hat{\mu}_i) \\
&= -\frac{1}{N}\sum_{i=1}^{N} y_i \ln(\hat{\mu}_i) + (1-y_i)\ln(1-\hat{\mu}_i) \\
&= -\frac{1}{N}\sum_{i=1}^{N} y_i \ln\Big(\text{logit}^{-1}\Big(x_i^T\hat{\beta}\Big)\Big) + (1-y_i)\ln\Big(1-\text{logit}^{-1}\Big(x_i^T\hat{\beta}\Big)\Big)
\end{aligned}$$

where we've used $\hat{\mu}_i = \text{logit}^{-1}\Big(x_i^T\hat{\beta}\Big)$. Doing this for both the training and the test data we get:

```
Training Data Logistic Loss 0.599063453561973
Test Data Logistic Loss 0.550330941693987
```

This value is quite small which suggests the model that we have created is quite good, and the fact that the logistic loss for both the training data and the test data are approximately equal suggests that the model hasn't overfit to the training data.

Question 3

To find the confidence interval for $\mathbb{P}$(Roger Federer beats Andy Murray), we note from the model that

$$\mathbb{P}(\text{Roger Federer beats Andy Murray}) = \text{logit}^{-1}\big(x^T\beta\big)$$

where x is a vector where the entry for Roger Federer is 1, the entry for Andy Murray is -1, and all other entries are 0, therefore to find a 68% confidence interval for $\mathbb{P}$(Roger Federer beats Andy Murray), we find a 68% confidence interval for $x^T\beta$ and plug into the above.

To find a 68% confidence interval for $x^T\beta$, we use the fact that the Maximum Likelihood Estimator $\hat{\beta}$ has asymptotic distribution

$$\hat{\beta} \sim AN_p\Big(\beta,\ i_{\beta}^{-1}(\beta,\sigma^2)\Big)$$

where, $i_\beta\big(\beta,\sigma^2\big)$ is the fisher information, for the parameters of β, of the generalised linear model, so

$$x^T\hat{\beta} \sim AN_p\Big(x^T\beta,\ x^Ti_\beta^{-1}\big(\beta,\sigma^2\big)x\Big)$$

This gives 68% confidence interval for β of

$$x^T\beta \in \Big[x^T\hat{\beta} - z\sqrt{x^Ti_\beta^{-1}(\beta,\sigma^2)x},\ x^T\hat{\beta} + z\sqrt{x^Ti_\beta^{-1}(\beta,\sigma^2)x}\Big] \tag{3}$$

where z satisfies $\mathbb{P}(|Z| \leq z) = 0.68$, $Z \sim N(0,\ 1)$. For a Bernoulli generalised linear model we may approximate $i_\beta\big(\beta,\sigma^2\big)$ using X^TWX where,

$$W = \begin{pmatrix} \hat{\mu}_1(1-\hat{\mu}_1) & 0 & \cdots & 0 \\ 0 & \hat{\mu}_1(1-\hat{\mu}_1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\mu}_n(1-\hat{\mu}_n) \end{pmatrix}$$

and $\hat{\mu}_i = \text{logit}^{-1}\Big(x_i^T\hat{\beta}\Big)$, putting this all into Equation 3, we get a 68% confidence interval for $x^T\beta$ of

$$x^T\beta \in \Big[x^T\hat{\beta} - z\sqrt{x^TX^TWXx},\ x^T\hat{\beta} + z\sqrt{x^TX^TWXx}\Big]$$

thus a 68% confidence interval for $\mathbb{P}$(Roger Federer beats Andy Murray) of

$$\begin{aligned}\mathbb{P}(\text{Roger Federer beats Andy Murray}) \in \Big[&\text{logit}^{-1}\Big(x^T\hat{\beta} - z\sqrt{x^TX^TWXx}\Big), \\ &\text{logit}^{-1}\Big(x^T\hat{\beta} + z\sqrt{x^TX^TWXx}\Big)\Big]\end{aligned}$$

Performing this calculation we get confidence interval

```
Confidence interval the for probability that Roger Federer beats Andy Murray is
[0.612491428102162, 0.690861133609206]
```

Question 4

For this model we will fix all the coefficients for the player "Agassi A." at 0, and will also fix all coefficients relating to performance on hard courts at 0[1]. Doing this we find the coefficients shown in Appendix A.1.2

Now calculating the logistic loss using the training data and the test data we get

```
Training Data Logistic Loss 0.574366693806463
Test Data Logistic Loss 0.852358259683162
```

[1]We could instead directly consider constraint that the average of $\beta_{a,s}$ over all surfaces is 0, s, for any fixed player a. This would allow for easier interpretation of all of the coefficients (β_a would be average performance of the player, and $\beta_{a,s}$ is the relative advantage/disadvantage a player gets when playing on surface s), but would require more complex computation (would require minimisation with constraints, so would have to use a method like Lagrange multipliers, which require more computational power).

We should note that although directly the constraint on the average of $\beta_{a,s}$ may cause difficulties, it can be deduced by using the constraint on coefficients relating to performance on hard courts by noting the model is invariant under transformation $\beta_a \to \beta_a + c_a$, $\beta_{a,s} \to \beta_{a,s} - c_a$, where c_a is allowed to vary from player to player. Thus if we first perform calculation with constraint on coefficients relating to performance on hard courts, then use the above transformation with c_a as average of $\beta_{a,s}$, we get the result with the constraint that the average of $\beta_{a,s}$ over all surfaces is 0.

This suggests that although the model is doing well on the training data, it may be over-fitting on this data, leading to a significantly worse performance on the test data. This is most likely due the fact that we have quadrupled the size of the parameter space, but have not changed the amount of training data we use, so the model can more easily fall into the trap of over-fitting.

Question 5

We can consider the model proposed in Question 2 to be contained in the model proposed in Question 4, thus, writing $\beta = (\beta_0,\ \beta_1)$, where β_1 represents the parameters to do with the surface of play, we may consider a hypothesis test of the form $H_0 : \beta_1 = 0$ against $H_1 : \beta_1 \neq 0$. H_0 represents the model presented in Question 2, and H_1 represents the model presented in Question 4.

Let $B = \mathbb{R}^p$, $B_0 = \{\beta \in B \mid \beta_1 = 0\}$.We consider the quantity

$$\Lambda(H_0) = 2\log\left\{\frac{\sup_{\beta' \in B} L(\beta')}{\sup_{\beta' \in B_0} L(\beta')}\right\} = 2\left\{\sup_{\beta' \in B} \ell\big(\beta'\big) - \sup_{\beta' \in B_0} \ell\big(\beta'\big)\right\}$$

Where L is the likelihood function and ℓ is the log-likelihood function, by Wilks' theorem,

$$\Lambda(H_0) \xrightarrow{d} \chi^2_k$$

where $k = \dim(B) - \dim(B_0) = p - p_0$. p denotes the number of parameters in the model presented in Question 4, and p_0 denotes the number of parameters in the model presented in Question 2. Let $\hat{\beta}_0$, $\hat{\beta}_1$ be the MLE of β in the models represented by H_0, H_1 respectively, then

$$\sup_{\beta' \in B_0} \ell\big(\beta'\big) = \ell\Big(\hat{\beta}_0\Big)$$
$$\sup_{\beta' \in B} \ell\big(\beta'\big) = \ell\Big(\hat{\beta}_1\Big)$$

so

$$\Lambda(H_0) = 2\Big(\ell\Big(\hat{\beta}_1\Big) - \ell\Big(\hat{\beta}_0\Big)\Big)$$

This can easily be calculated, and the p-value can be found using $\chi^2_{p-p_0}$, This gives the results

```
Test Statistic: 221.628722045949871
P-value for the likelihood ratio test: 0.005518082704732
```

This suggests that the model created in Question 4 is significantly better than the model suggested in Question 2, and we can reject the simpler model at the 1% level. This does not agree with our cross-validation as we saw that the simpler model performed much better on the unseen test data, than the more complex model, suggesting that the more complex model has over-fit to the training data.

Question 6

We can use the variables `W1-W5` and `L1-L5` to give us more information on the precise number of games each player won in a given match, thus (hopefully) allowing our model to more accurately predict the number of matches by either

- Model for each game individually instead of for each match

  From this we would get an estimate of the probability that a player wins a game, and from this derive the probability that the player wins the match given the probability they win a game

- Assume $\mathbb{P}(\text{Player } a \text{ wins a match against Player } b) \approx \mathbb{P}(\text{Player } a \text{ wins a game against Player } b)$

  This would allow us to just assume that the number of games where player a beats player b, is approximately equal to the number of matches where player a beat player b. This gives us more data to work with, and thus (hopefully) a more accurate model.

Regularisation

Question 7

When the function `glmnet` standardises a design matrix, it transforms the matrix in a way such that for each $j = 1, \dots, p$

$$\sum_{i=1}^{N} X_{ij} = 0$$

and

$$\sqrt{\sum_{i=1}^{N} \frac{X_{ij}^2}{N}} = 1$$

i.e. each column of the design matrix has mean 0 and standard deviation 1. This makes sense if the data is not all of the same scale, as data at larger scales are likely to dominate the fit of the model. In our model, the design matrix already has all of the columns at the correct scale, so we do not need to standardise our model.

When we use the model we get the result shown in Appendix A.2.1, and we find the optimal value of λ is

```
Lambda which minimises the mean cross-validated error is: 0.000209427079577062
```

Question 8

With the optimal value of λ, we find that the number of estimates for $\beta_{a,s}$ that are non-zero is

```
The number of non zero surface terms are: 72
```

and calculating the logistic loss for both the training and the test data we get

```
Training Data Logistic Loss 0.576729176439859
Test Data Logistic Loss 0.555683332673685
```

The logistic loss on the training for this model is slightly larger than the logistic loss for the model proposed in Question 4, however the logistic loss on the test data is significantly smaller for this model compared to the model proposed in Question 4. These results suggest that the new model generalises better than the model proposed in Question 4, i.e. adding regularisation has allowed us to generate a model which generalises much better.

Question 9

We are going to change the model suggested in Question 2 to assume that the performance of each player changes linearly over time[2], so we can introduce a parameter $\beta_{a,p}$ to be the change in performance per year of player a. Our new model thus becomes

$$\mathbb{P}(\text{Player } a \text{ wins a match against Player } b \text{ in year } t) = \frac{\exp(\beta_a + t\beta_{a,p} - \beta_b - t\beta_{b,p})}{1 + \exp(\beta_a + t\beta_{a,p} - \beta_b - t\beta_{b,p})}$$

thus the rows of our design matrix are similar to the design matrix for the model of Question 2, with the addition that the entry representing $\beta_{a,p}$ is t and the entry representing $\beta_{b,p}$ is $-t$ (all other entries

[2] We could instead consider a model where we add parameters for each year, for each player, but if we do this we face two problems

1. There's a high probability the model will over-fit to the training data provided
2. We would need training data from all the years played

   This would prevent the model from making any predictions about future matches(and also means if we wanted to compare the model to those already suggested, we would have to retrain all of those models on the new training data) are 0 as before). In order to make sure the model isn't dominated by the terms $\beta_{a,p}$ in training, we will map the years 2000-2014 to $t \in [0, 1]$, we will also use the constraint that all coefficients relating to player "Agassi A." are 0. We will consider two models of this form,

1. A model with no weights
2. A model with weights exclusively on the $\beta_{a,p}$ coefficients

Doing this we get the coefficients shown in Appendix A.2.2 and Appendix A.2.3 respectively and the Logistic Losses

```
Training Data Logistic Loss 0.571284100667494
Test Data Logistic Loss 0.548636266336129
```

```
Training Data Logistic Loss 0.572748391872170
Test Data Logistic Loss 0.554890223106983
```

Suggesting similar performance, although the model with the lasso penalty performances slightly worse in both measures.

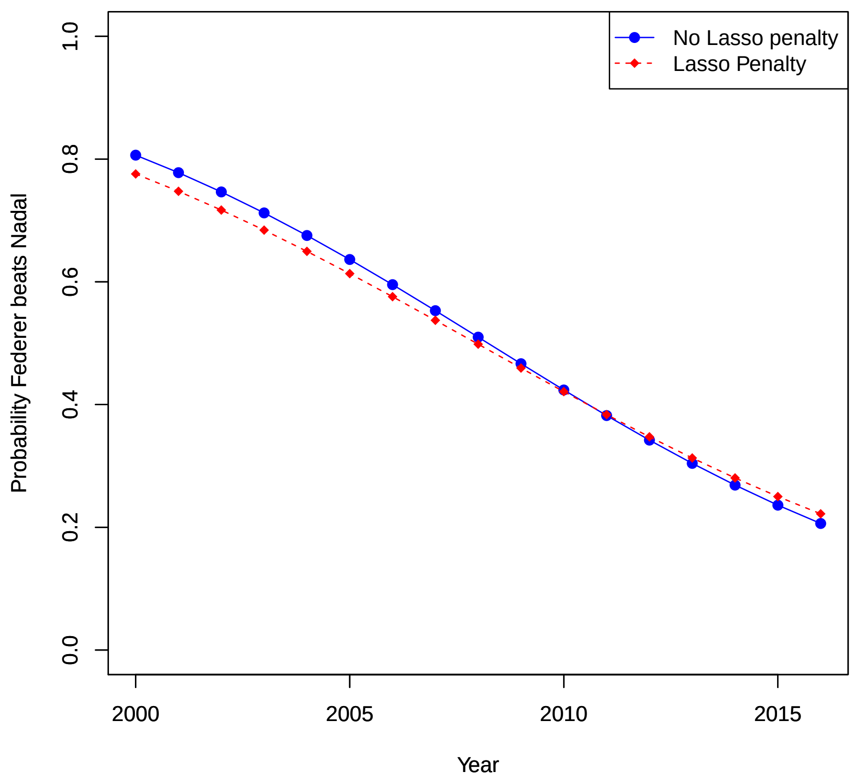

Figure 1: Graph showing the probability that Roger Federer beats Rafael Nadal against time

We can see in Figure 1 that as time continues onwards, the probability that Federer beats Nadal decreases.

To compare all of the models, we will look at the Logistic Loss and we will use the Akaike Information Criterion (AIC) to approximate the Kullback-Leibler Divergence (KLD) of all the models using the fact that the KLD is approximately equal to AIC$/2n$ where n is the number of samples used and n large. We will label the models as follows

- **Model 1**: The model presented in question 2
- **Model 2**: The model presented in question 4

- **Model 3**: The model presented in question 7
- **Model 4**: The model presented in question 9 without the lasso penalty
- **Model 5**: The model presented in question 9 with the lasso penalty

As we require n to be large in our approximation of the KLD, we will only approximate the KLD using the training data. Doing this we get the following results

```
Model 1
KB: Training Set: 0.61221255095444
LL: Training Set: 0.599063453561973 Test Set: 0.550330941693987
Model 2
KB: Training Set: 0.625625887031335
LL: Training Set: 0.574366693806463 Test Set: 0.852358259683162
Model 3
KB: Training Set: 0.62798836966473
LL: Training Set: 0.576729176439859 Test Set: 0.555683332673685
Model 4
KB: Training Set: 0.597582295452428
LL: Training Set: 0.571284100667494 Test Set: 0.548636266336129
Model 5
KB: Training Set: 0.599046586657104
LL: Training Set: 0.57274839187217 Test Set: 0.554890223106983
```

This suggests that out of all the models we have proposed, model 4 is the best with respect to all the measures considered above.

Can you outperform the betting market?

Question 10

Let $B_{i,W}$ be the betting odds for the winning player of the i^{th} match and let $B_{i,L}$ be defined similarly for the losing player. Then we can say

$$\begin{aligned} R_{i,W} &\sim \phi_{i,W}(B_{i,W}-1)Y_i \\ R_{i,L} &\sim \phi_{i,L}(B_{i,L}-1)(1-Y_i) \end{aligned}$$

where Y_i are independent Bernoulli variables (implying pairs $(R_{i,L}, R_{i,W})$ are independent for different values of i) with

$$\mathbb{P}(Y_i = 1) = \mathbb{P}(\text{Player } W_i \text{ wins a match against Player } L_i) =: \mu_{i,W}$$

W_i is winner of i^{th} match, L_i is loser of the i^{th} match, so we can write $Y_i \sim \text{Ber}(\mu_{i,W})$.

Thus

$$\begin{aligned} \mathbb{E}[R] &= \mathbb{E}\left[\sum_{i=1}^{k}(R_{i,W}+R_{i,L})\right] \\ &= \sum_{i=1}^{k}(\mathbb{E}[R_{i,W}]+\mathbb{E}[R_{i,L}]) \\ &= \sum_{i=1}^{k}\phi_{i,W}(B_{i,W}-1)\mathbb{E}[Y_i]+\phi_{i,L}(B_{i,L}-1)(1-\mathbb{E}[Y_i]) \\ &= \sum_{i=1}^{k}\phi_{i,W}(B_{i,W}-1)\mu_{i,W}+\phi_{i,L}(B_{i,L}-1)(1-\mu_{i,W}) \\ &= \omega^T q \end{aligned}$$

where $\omega = \phi$ and q is a vector such that

$$\begin{aligned} q_{i,W} &= (B_{i,W} - 1)\mu_{i,W} \\ q_{i,L} &= (B_{i,L} - 1)(1 - \mu_{i,W}) \end{aligned}$$

We also have

$$\begin{aligned}
\nu \mathrm{Var}(R) &= \nu \mathrm{Var}\left(\sum_{i=1}^{k} (R_{i,W} + R_{i,L})\right) \\
&= \nu \sum_{i=1}^{k} \mathrm{Var}(R_{i,W} + R_{i,L}) \\
&= \nu \sum_{i=1}^{k} \mathrm{Var}(\phi_{i,W}(B_{i,W} - 1)Y_i + \phi_{i,L}(B_{i,L} - 1)(1 - Y_i)) \\
&= \nu \sum_{i=1}^{k} \begin{pmatrix} \phi_{i,W}(B_{i,W} - 1), & \phi_{i,L}(B_{i,L} - 1) \end{pmatrix} \mathrm{Cov}\begin{pmatrix} Y_i \\ 1 - Y_i \end{pmatrix} \begin{pmatrix} \phi_{i,W}(B_{i,W} - 1) \\ \phi_{i,L}(B_{i,L} - 1) \end{pmatrix} \\
&= \sum_{i=1}^{k} \begin{pmatrix} \phi_{i,W}, & \phi_{i,L} \end{pmatrix} \nu A_i \begin{pmatrix} \phi_{i,W} \\ \phi_{i,L} \end{pmatrix} \\
&= \omega^T Q \omega
\end{aligned}$$

where $\omega = \phi$ and Q is a matrix such that

$$Q = \nu \begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_k \end{pmatrix}$$

A_i is a matrix

$$\begin{aligned}
A_i &= \mu_{i,W}(1 - \mu_{i,W}) \begin{pmatrix} B_{i,W} - 1 & 0 \\ 0 & B_{i,L} - 1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} B_{i,W} - 1 & 0 \\ 0 & B_{i,L} - 1 \end{pmatrix} \\
&= \mu_{i,W}(1 - \mu_{i,W}) \begin{pmatrix} (B_{i,W} - 1)^2 & -(B_{i,W} - 1)(B_{i,L} - 1) \\ -(B_{i,W} - 1)(B_{i,L} - 1) & (B_{i,L} - 1)^2 \end{pmatrix}
\end{aligned}$$

So given $B_{i,W}, B_{i,L} \geq 1 \quad \forall i = 1, \ldots, k$, then A_i positive semi-definite for all $i = 1, \ldots, k$, thus Q positive semi-definite. Note that the conditions

$$\phi_{i,W}, \phi_{i,L} \geq 0 \text{ for } i = 1, \ldots, k$$
$$\sum_{i=1}^{k} \phi_{i,W} + \phi_{i,L} = 1000$$

imply

$$\omega_i \geq 0 \text{ for } i = 1, \ldots, 2k$$
$$\omega^T \mathbf{1} = 1000$$

thus the problem

$$\underset{\phi \in \mathbb{R}^{2k}}{\text{minimise}} \quad -\mathbb{E}[R] + \nu \mathrm{Var}(R)$$

$$\text{subject to } \phi_{i,W} \geq 0, \phi_{i,L} \geq 0 \text{ for } i = 1, \ldots, k, \ \sum_{i=1}^{k} [\phi_{i,W} + \phi_{i,L}] = 1000$$

can be written as

$$\begin{aligned} &\underset{\omega \in \mathbb{R}^{2k}}{\text{minimise}} \quad \omega^T Q \omega - \omega^T q \\ &\text{subject to } \omega^T \mathbf{1} = 1000, \ \omega_i \geq 0 \text{ for } i = 1, \ldots, 2k \end{aligned}$$

where q, Q are written in terms of $B_{i,W}$, $B_{i,L}$ which are given to us, and $\mu_{i,W}$ which can be predicted from a logistic regression model.

Question 11

The code to find the optimal portfolio can be found in the program Listings section.

For the total profit, I considered both Models 2 and 3, the results of which can be found in Figure 2.

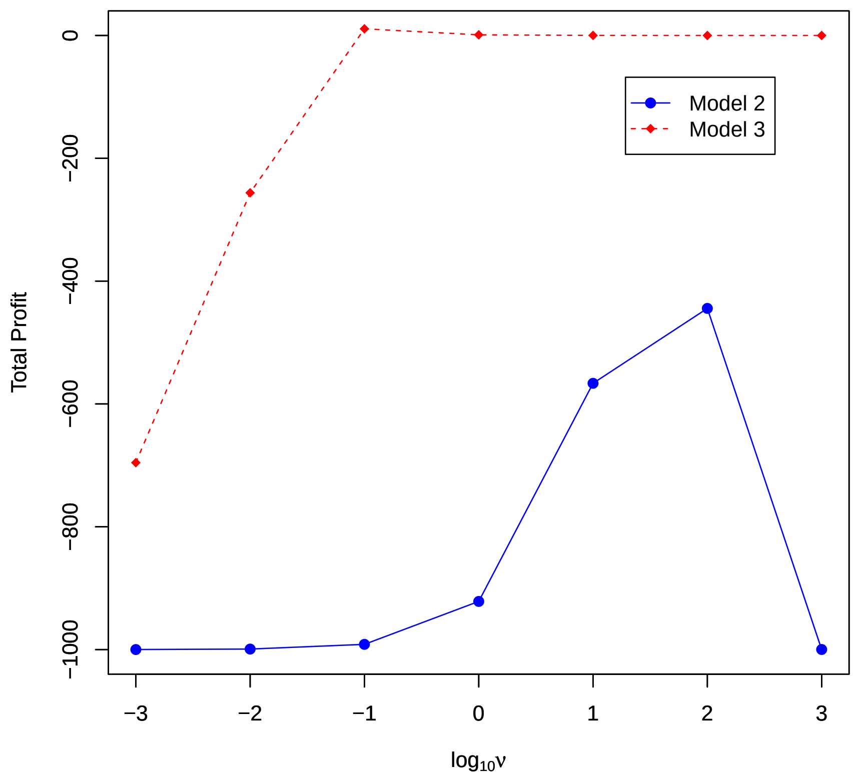

Figure 2: Graph showing the total profit made from the optimal Markowitz Portfolio for Models 2 and 3

We can see that for all values of ν that the predictions from Model 3 seem to outperform the predictions from Model 2, which is to be expected as we found that Model 3 seemed to generalise better than Model 2. For model 2, we see that from $\nu = 10^{-3}$ to $\nu = 10^2$ the total profit increases, this suggests that for small values of ν the money is being mainly placed on risky bets and is then lost. We also see that as ν increases past $\nu = 10^2$ the total profit quickly decreases. So for Model 2, we see that the optimal value of ν to use is $\nu = 10^2$

Question 12

Looking through our argument in Question 10, we can see that we assumed to know the exact value of $\mathbb{E}[Y_i]$ in order to have an easily written down form for the vector q and the matrix Q, so we can simply write down the more general form of q, and Q

The $q \in \mathbb{R}^{2k}$ is such that

$$\begin{aligned} q_{i,W} &= (B_{i,W} - 1)\mathbb{E}[Y_i] \\ q_{i,L} &= (B_{i,L} - 1)(1 - \mathbb{E}[Y_i]) \end{aligned}$$

where $\mathbb{E}[Y_i]$ calculated using posterior distribution of $\mu_{i,W}$,

The matrix Q is such that

$$Q = \nu \begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_k \end{pmatrix}$$

where A_i are matrices such that

$$A_i = \begin{pmatrix} B_{i,W} - 1 & 0 \\ 0 & B_{i,L} - 1 \end{pmatrix} \mathrm{Cov} \begin{pmatrix} Y_i \\ 1 - Y_i \end{pmatrix} \begin{pmatrix} B_{i,W} - 1 & 0 \\ 0 & B_{i,L} - 1 \end{pmatrix}$$

where $\mathrm{Cov}(Y_i, 1 - Y_i)$ again to be calculated using the posterior distribution of $\mu_{i,W}$. This leaves us with a quadratic optimisation problem with affine constraints.

Question 13

Assume that before the i^{th} bet, we have a total bankroll of

$$b_i,$$

then after this bet, we would have a total bankroll of

$$\begin{aligned} R_i &\coloneqq b_i(1 - (\phi_{i,W} + \phi_{i,L})) + b_i\phi_{i,W}B_{i,W}Y_i + b_i\phi_{i,L}B_{i,L}(1 - Y_i) \\ &= b_i((1 - (\phi_{i,W} + \phi_{i,L})) + \phi_{i,W}B_{i,W}Y_i + \phi_{i,L}B_{i,L}(1 - Y_i)) \end{aligned}$$

where Y_i is defined the same as in Question 10. For the Kelly Criterion we want to maximise

$$\begin{aligned} \mathbb{E}[\log R_i] &= \mathbb{E}[\log(b_i((1 - (\phi_{i,W} + \phi_{i,L})) + \phi_{i,W}B_{i,W}Y_i + \phi_{i,L}B_{i,L}(1 - Y_i)))] \\ &= \log b_i + \mu_{i,W}\log(1 - \phi_{i,L} + \phi_{i,W}(B_{i,W} - 1)) \\ &\qquad + (1 - \mu_{i,W})\log(1 - \phi_{i,W} + \phi_{i,L}(B_{i,L} - 1)) \end{aligned}$$

which is equivalent to maximising

$$\mu_{i,W}\log(1 - \phi_{i,L} + \phi_{i,W}(B_{i,W} - 1)) + (1 - \mu_{i,W})\log(1 - \phi_{i,W} + \phi_{i,L}(B_{i,L} - 1)).$$

We can approximate $\mu_{i,W}$ using the logistic models defined above, and thus find approximate solutions to the above maximisation problem. Doing this using Model 2 to find approximations from $\mu_{i,W}$, we get the graph shown in Figure 3

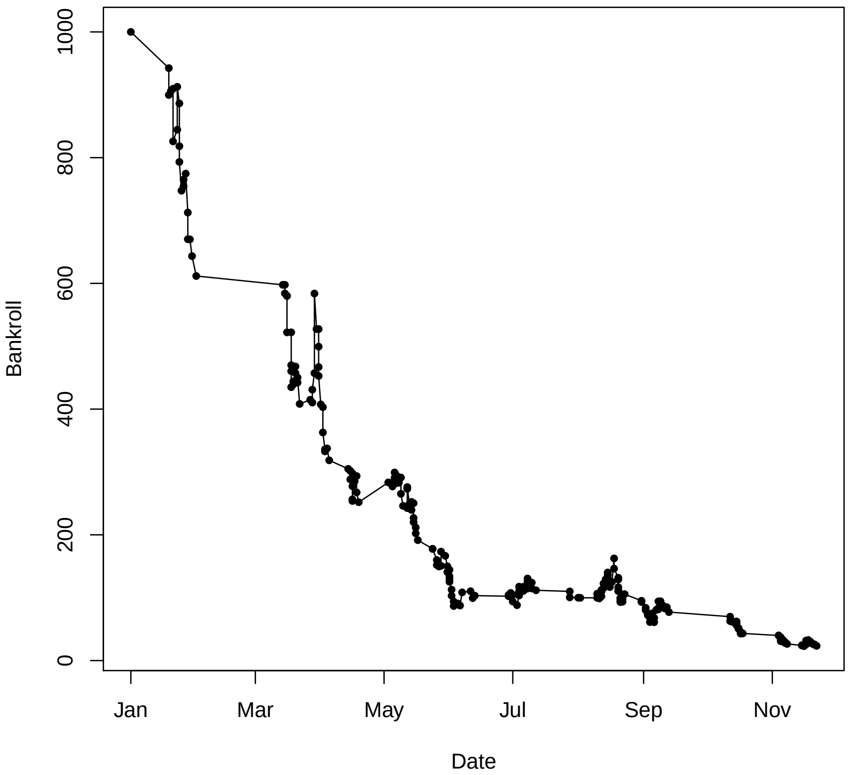

Figure 3: Graph showing the Bankroll over time using the Kelly Fractions generated using approximations from Model 2

Appendices

A Large Program Outputs

A.1 No Regularisation

A.1.1 Coefficients for the Generalised Linear Model Presented in Question 2

```
Almagro N. : -1.17952567977698
Anderson K. : -1.41740048718148
Baghdatis M. : -0.945474430637639
Benneteau J. : -1.35829168415513
Berdych T. : -0.495500995000218
Blake J. : -0.943495297113422
Canas G. : -0.491947794379522
Chardy J. : -1.7402143728838
Chela J.I. : -1.23372743360869
Cilic M. : -1.08010438567443
Clement A. : -1.47790403980301
Coria G. : -0.131876751656099
Davydenko N. : -0.569553964807553
Del Potro J.M. : -0.0887866744546049
Djokovic N. : 0.867885326283429
Federer R. : 0.863596466168205
Ferrer D. : -0.352304771357835
Ferrero J.C. : -0.495425976375771
Fish M. : -0.887968211203663
Fognini F. : -1.58415369171794
Gasquet R. : -0.717995002779436
Gaudio G. : -0.810410147274934
Gonzalez F. : -0.664136513214158
Grosjean S. : -1.16963812733477
Haas T. : -0.790354609101343
Henman T. : -0.720401118904515
Hewitt L. : -0.167282021983178
Isner J. : -1.06302445658751
Karlovic I. : -1.16266666421072
Kiefer N. : -0.946826957387466
Kohlschreiber P. : -1.39210552832257
Kuerten G. : -0.278257813168006
Ljubicic I. : -0.769727612089247
Lopez F. : -1.21707368970697
Mathieu P.H. : -1.326728951728
Melzer J. : -1.225441336725
Monaco J. : -1.16026352918692
Monfils G. : -0.733540043095985
Moya C. : -0.774249785099166
Murray A. : 0.232623848540625
Nadal R. : 1.11150911110391
Nalbandian D. : -0.335734681453467
Nieminen J. : -1.63385319304061
Nishikori K. : -0.642282001055427
Novak J. : -1.20671475341909
Raonic M. : -0.827827018929102
Robredo T. : -0.925413948934426
Roddick A. : -0.127424702440009
Safin M. : -0.689529811826729
Schuettler R. : -1.3975084708622
Seppi A. : -1.55913125573001
Simon G. : -0.884264344389769
Soderling R. : -0.412814969287601
Stepanek R. : -1.03424863740179
Tipsarevic J. : -1.24851947156759
Tsonga J.W. : -0.203353852215378
Verdasco F. : -0.92311096777289
Wawrinka S. : -0.41410877604298
Youzhny M. : -1.0450496018041
```

A.1.2 Coefficients for the Generalised Linear Model Presented in Question 4

```
Almagro N.: -1.3892082146162
Almagro N. Carpet: -17.349869558802
Almagro N. Clay: 1.00416712413898
Almagro N. Grass: -1.57309395808088
Anderson K.: -1.26221395269597
Anderson K. Clay: -0.0339725810476121
Anderson K. Grass: -2.59122641337264
Baghdatis M.: -0.910092398683705
Baghdatis M. Carpet: 3.62902306178462
Baghdatis M. Clay: -0.6419868260073
Baghdatis M. Grass: -0.0606850012518071
Benneteau J.: -1.26138240855153
Benneteau J. Carpet: 0
Benneteau J. Clay: 0.203545184427564
Benneteau J. Grass: -1.93586955755485
Berdych T.: -0.601911099910022
Berdych T. Carpet: 2.49668251476031
Berdych T. Clay: 0.657338628381247
Berdych T. Grass: -0.900818634880525
Blake J.: -0.798109978864451
Blake J. Carpet: 0.153466897129021
Blake J. Clay: 0.0421919225264755
Blake J. Grass: 0
Canas G.: -0.421998083376412
```

Canas G. Carpet: 1.81953394760265
Canas G. Clay: 0.467570293631678
Canas G. Grass: -21.8566766324771
Chardy J.: -1.38434884476345
Chardy J. Carpet: -19.9993508211936
Chardy J. Clay: -0.0189965279717889
Chardy J. Grass: -43.2199449832743
Chela J.I.: -1.10869978707149
Chela J.I. Carpet: 0
Chela J.I. Clay: 0.192206492083836
Chela J.I. Grass: -23.2130872578698
Cilic M.: -0.808513672613375
Cilic M. Carpet: -20.824477360029
Cilic M. Clay: -0.945611597347284
Cilic M. Grass: -1.08632821148281
Clement A.: -1.48361130207282
Clement A. Carpet: 0.136815397893801
Clement A. Clay: 0.166371238854093
Clement A. Grass: -0.3468800868369
Coria G.: -0.998093473504951
Coria G. Carpet: 1.93431346549853
Coria G. Clay: 2.25057511411051
Coria G. Grass: 0.044016484241454
Davydenko N.: -0.450174512299637
Davydenko N. Carpet: 2.52877092618563
Davydenko N. Clay: 0.27951761594741
Davydenko N. Grass: -2.60298057044645
Del Potro J.M.: -0.0535508487471832
Del Potro J.M. Carpet: 0.803839967282386
Del Potro J.M. Clay: 0.484056312146409
Del Potro J.M. Grass: -1.6711876061441
Djokovic N.: 0.942923475961583
Djokovic N. Carpet: -41.6642349945601
Djokovic N. Clay: 0.690488514371589
Djokovic N. Grass: -0.458007013590508
Federer R.: 0.797956965336493
Federer R. Carpet: -1.68547435457592
Federer R. Clay: 0.945468408456234
Federer R. Grass: 0.913164669941428
Ferrer D.: -0.465164972997128
Ferrer D. Carpet: 1.07031049369319
Ferrer D. Clay: 1.09102234873366
Ferrer D. Grass: -1.57095596870208
Ferrero J.C.: -0.880698914157015
Ferrero J.C. Carpet: 0.620681431670479
Ferrero J.C. Clay: 1.57193989574283
Ferrero J.C. Grass: -0.675895491585469
Fish M.: -0.726002087067941
Fish M. Carpet: 0.604347887153206
Fish M. Clay: -1.60868570081817
Fish M. Grass: -0.523595239495463
Fognini F.: -1.86457911617291
Fognini F. Clay: 1.10941925941738
Fognini F. Grass: -2.30818416938278
Gasquet R.: -0.873235628346689
Gasquet R. Carpet: 1.9042024543693
Gasquet R. Clay: 0.750839868595031
Gasquet R. Grass: 0.752927624716194
Gaudio G.: -1.08141171069086
Gaudio G. Carpet: 0.167479266411282
Gaudio G. Clay: 1.31719930804762
Gaudio G. Grass: -42.8139048106458
Gonzalez F.: -0.629259121756498
Gonzalez F. Carpet: -40.092052396954
Gonzalez F. Clay: 0.858241223547606
Gonzalez F. Grass: -2.07593370197475
Grosjean S.: -1.42527931478905
Grosjean S. Carpet: 1.489467009293
Grosjean S. Clay: 0.893362789571874
Grosjean S. Grass: 0.835171718488097
Haas T.: -0.696247175958088
Haas T. Carpet: 0.906232217215298
Haas T. Clay: -0.00709124418784257
Haas T. Grass: -0.848540400481158
Henman T.: -0.9663744459613
Henman T. Carpet: 3.78962747769529
Henman T. Clay: 0.496573678731571
Henman T. Grass: 0.793024065287348
Hewitt L.: -0.205276609191986
Hewitt L. Carpet: 1.7550287938701
Hewitt L. Clay: 0.42013559782029
Hewitt L. Grass: -0.474380363206423
Isner J.: -0.829682654331771
Isner J. Clay: -1.25243347465575
Isner J. Grass: -2.28306392732966
Karlovic I.: -1.12619855138838
Karlovic I. Carpet: -1.32249667006711
Karlovic I. Clay: -0.175920789262609
Karlovic I. Grass: 1.04048374645746
Kiefer N.: -0.738742959810925
Kiefer N. Carpet: -1.82677537599099
Kiefer N. Clay: -0.0800242814109295
Kiefer N. Grass: -1.52738425047479
Kohlschreiber P.: -1.44925045737678
Kohlschreiber P. Carpet: 2.45930284444152
Kohlschreiber P. Clay: 0.626122881254295
Kohlschreiber P. Grass: -1.70189240003281
Kuerten G.: -0.626968281399595
Kuerten G. Carpet: 1.13225297411952
Kuerten G. Clay: 1.38917101000105
Ljubicic I.: -0.85416319926596
Ljubicic I. Carpet: 1.68481644805926
Ljubicic I. Clay: 0.881086898827829
Ljubicic I. Grass: -1.45481508336375
Lopez F.: -1.23275498410422
Lopez F. Carpet: 1.21042452785876

Lopez F. Clay: 0.119396386607115
Lopez F. Grass: 0.028351449761682
Mathieu P.H.: -1.25089351197653
Mathieu P.H. Carpet: -18.6031067770127
Mathieu P.H. Clay: 0.0439628804604372
Mathieu P.H. Grass: -1.1814701735193
Melzer J.: -1.40293809458837
Melzer J. Carpet: 2.99302103358606
Melzer J. Clay: 0.661508276410367
Melzer J. Grass: 0.0419519802357465
Monaco J.: -1.34091644707745
Monaco J. Carpet: 1.03582167809778
Monaco J. Clay: 1.01274570010788
Monaco J. Grass: -22.3699926169893
Monfils G.: -0.884481625203574
Monfils G. Carpet: 1.20781344464074
Monfils G. Clay: 0.967106502922535
Monfils G. Grass: -23.8743824600551
Moya C.: -1.01625706897567
Moya C. Carpet: 1.93009061571549
Moya C. Clay: 1.17337393084969
Moya C. Grass: -1.35770404286955
Murray A.: 0.346728628690948
Murray A. Carpet: -0.0626070680569486
Murray A. Clay: -0.167894058045759
Murray A. Grass: 0.30048551532211
Nadal R.: 0.533721389621136
Nadal R. Carpet: 1.7720528278484
Nadal R. Clay: 2.41311155076404
Nadal R. Grass: 1.37661412090141
Nalbandian D.: -0.423642795697449
Nalbandian D. Carpet: 3.30121801204521
Nalbandian D. Clay: 0.380014513644419
Nalbandian D. Grass: -0.960137551892846
Nieminen J.: -1.90090498074066
Nieminen J. Carpet: 1.12790395412209
Nieminen J. Clay: 1.15724721069075
Nieminen J. Grass: -0.841154722556591
Nishikori K.: -0.575571587439478
Nishikori K. Clay: 1.19165491068368
Nishikori K. Grass: -39.0278321572598
Novak J.: -1.44237520306713
Novak J. Carpet: 1.14413587903551
Novak J. Clay: 0.82480066999246
Novak J. Grass: 21.0228960668504
Raonic M.: -0.649825748757006
Raonic M. Clay: -0.0588802086338639
Raonic M. Grass: -18.1363882059322
Robredo T.: -1.1097618084833
Robredo T. Carpet: 1.3427159786612
Robredo T. Clay: 0.846919376306443
Robredo T. Grass: 0
Roddick A.: -0.120392929315066
Roddick A. Carpet: 1.36757267829993
Roddick A. Clay: -0.0631109266471156
Roddick A. Grass: -0.426785240554999
Safin M.: -0.878432103600142
Safin M. Carpet: 3.82564585192518
Safin M. Clay: 0.400954614645235
Safin M. Grass: 0.335294960041612
Schuettler R.: -1.51624980940149
Schuettler R. Carpet: 0.103904069392326
Schuettler R. Clay: 0.432863852894716
Schuettler R. Grass: 0.506044177974829
Seppi A.: -1.86817872303262
Seppi A. Carpet: 0
Seppi A. Clay: 1.21537447781998
Seppi A. Grass: -1.44291957758771
Simon G.: -0.798934441137798
Simon G. Carpet: 0
Simon G. Clay: 0.507916419705139
Simon G. Grass: -1.91082804848761
Soderling R.: -0.323685165799524
Soderling R. Carpet: -22.2018510427716
Soderling R. Clay: 0.538545200515955
Soderling R. Grass: -0.542361963022721
Stepanek R.: -1.23629409509337
Stepanek R. Carpet: 2.79505028381474
Stepanek R. Clay: 0.897935871409348
Stepanek R. Grass: -1.30390987780418
Tipsarevic J.: -1.18420386238635
Tipsarevic J. Carpet: 0
Tipsarevic J. Clay: 0.352617252498138
Tipsarevic J. Grass: -0.747018381746187
Tsonga J.W.: -0.138098882514546
Tsonga J.W. Carpet: 1.57131107744411
Tsonga J.W. Clay: 0.201845614597706
Tsonga J.W. Grass: -0.00227514438898373
Verdasco F.: -1.17787348583564
Verdasco F. Carpet: -0.410868919333049
Verdasco F. Clay: 1.19585328749366
Verdasco F. Grass: -0.949142501150319
Wawrinka S.: -0.456511720338268
Wawrinka S. Carpet: -1.24596171130363
Wawrinka S. Clay: 0.808661347743702
Wawrinka S. Grass: -0.627073324654356
Youzhny M.: -0.855256464104461
Youzhny M. Carpet: 1.62167449368239
Youzhny M. Clay: -0.123309456408109
Youzhny M. Grass: -1.09418598844716

A.2 Regularisation

A.2.1 Coefficients for the Generalised Linear Model Presented in Question 7

```
Almagro N.: -1.234681551175
Almagro N. Carpet: 0
Almagro N. Clay: 0.276672771773717
Almagro N. Grass: 0
Anderson K.: -1.35118003754035
Anderson K. Clay: -0.0569577096524396
Anderson K. Grass: 0
Baghdatis M.: -0.795839243853994
Baghdatis M. Carpet: 0.580788690955746
Baghdatis M. Clay: -0.849136265616186
Baghdatis M. Grass: 0.173686389095082
Benneteau J.: -1.31681811831289
Benneteau J. Carpet: 0
Benneteau J. Clay: 0
Benneteau J. Grass: 0
Berdych T.: -0.470729146344779
Berdych T. Carpet: 0.341468159352423
Berdych T. Clay: 0
Berdych T. Grass: 0
Blake J.: -0.880232490430669
Blake J. Carpet: 0
Blake J. Clay: -0.13965047586489
Blake J. Grass: -0.916892488786821
Canas G.: -0.411793088318112
Canas G. Carpet: 0
Canas G. Clay: 0
Canas G. Grass: 0
Chardy J.: -1.64200919174551
Chardy J. Carpet: 0
Chardy J. Clay: 0
Chardy J. Grass: -0.371837908306008
Chela J.I.: -1.21531821860481
Chela J.I. Carpet: -0.107792568669288
Chela J.I. Clay: 0
Chela J.I. Grass: 0
Cilic M.: -0.824398705179295
Cilic M. Carpet: 0
Cilic M. Clay: -1.02912836464224
Cilic M. Grass: 0
Clement A.: -1.43549136310403
Clement A. Carpet: 0
Clement A. Clay: 0
Clement A. Grass: 0
Coria G.: -0.755794016506391
Coria G. Carpet: 0
Coria G. Clay: 1.3843018133815
Coria G. Grass: 0
Davydenko N.: -0.426734409668856
Davydenko N. Carpet: 0.110449093650088
Davydenko N. Clay: -0.130437056124346
Davydenko N. Grass: 0
Del Potro J.M.: -0.0564982612040388
Del Potro J.M. Carpet: 0
Del Potro J.M. Clay: 0
Del Potro J.M. Grass: 0
Djokovic N.: 1.00538114806127
Djokovic N. Carpet: -2.06524617711647
Djokovic N. Clay: 0
Djokovic N. Grass: 0
Federer R.: 0.878867145859966
Federer R. Carpet: -0.941520429689676
Federer R. Clay: 0.238729076437215
Federer R. Grass: 0.899347141737547
Ferrer D.: -0.415187576034492
Ferrer D. Carpet: 0
Ferrer D. Clay: 0.47434207217355
Ferrer D. Grass: -0.11505607292365
Ferrero J.C.: -0.739093974906478
Ferrero J.C. Carpet: 0
Ferrero J.C. Clay: 0.840251158990963
Ferrero J.C. Grass: 0
Fish M.: -0.712829519292545
Fish M. Carpet: 0
Fish M. Clay: -1.30857820596281
Fish M. Grass: 0
Fognini F.: -1.72881371322061
Fognini F. Clay: 0.357396708405956
Fognini F. Grass: 0
Gasquet R.: -0.670560538985379
Gasquet R. Carpet: 0.111365952374415
Gasquet R. Clay: 0
Gasquet R. Grass: 0.481802303259191
Gaudio G.: -1.07417760748647
Gaudio G. Carpet: 0
Gaudio G. Clay: 0.729223694357028
Gaudio G. Grass: 0
Gonzalez F.: -0.614607063162656
Gonzalez F. Carpet: -1.50190006175514
Gonzalez F. Clay: 0.239202568879226
Gonzalez F. Grass: -0.295746898798162
Grosjean S.: -1.14857978428717
Grosjean S. Carpet: 0
Grosjean S. Clay: 0
Grosjean S. Grass: 0.369788391066479
Haas T.: -0.64746693870594
Haas T. Carpet: 0
Haas T. Clay: -0.347501188474911
Haas T. Grass: 0
Henman T.: -0.768181842834629
Henman T. Carpet: 1.02692779490565
```

Henman T. Clay: 0
Henman T. Grass: 0.515243024388166
Hewitt L.: -0.108426330803942
Hewitt L. Carpet: 0
Hewitt L. Clay: 0
Hewitt L. Grass: 0
Isner J.: -0.889074388464509
Isner J. Clay: -0.982678995122046
Isner J. Grass: 0
Karlovic I.: -1.06959464015449
Karlovic I. Carpet: 0
Karlovic I. Clay: -0.381311590930214
Karlovic I. Grass: 0.692202300732646
Kiefer N.: -0.824267119119353
Kiefer N. Carpet: 0
Kiefer N. Clay: -0.23385670218373
Kiefer N. Grass: 0
Kohlschreiber P.: -1.34321346748185
Kohlschreiber P. Carpet: 0
Kohlschreiber P. Clay: 0
Kohlschreiber P. Grass: 0
Kuerten G.: -0.412570815046872
Kuerten G. Carpet: 0
Kuerten G. Clay: 0.481310435223682
Ljubicic I.: -0.7427850433353
Ljubicic I. Carpet: 0
Ljubicic I. Clay: 0.1950151473716
Ljubicic I. Grass: -0.148442254992521
Lopez F.: -1.13813954837927
Lopez F. Carpet: 0
Lopez F. Clay: -0.278051300029379
Lopez F. Grass: 0.358484958900001
Mathieu P.H.: -1.22430223683945
Mathieu P.H. Carpet: 0
Mathieu P.H. Clay: -0.235792338468826
Mathieu P.H. Grass: 0
Melzer J.: -1.17999018841087
Melzer J. Carpet: 0
Melzer J. Clay: 0
Melzer J. Grass: 0.136659014370137
Monaco J.: -1.24055562652475
Monaco J. Carpet: 0
Monaco J. Clay: 0.337082550416446
Monaco J. Grass: 0
Monfils G.: -0.789108073457385
Monfils G. Carpet: 0
Monfils G. Clay: 0.289384731781324
Monfils G. Grass: 0
Moya C.: -0.868012255802792
Moya C. Carpet: 0
Moya C. Clay: 0.452973747346851
Moya C. Grass: 0
Murray A.: 0.385083605129751
Murray A. Carpet: 0
Murray A. Clay: -0.590022793142046
Murray A. Grass: 0.474745647086021
Nadal R.: 0.655622717493339
Nadal R. Carpet: 0
Nadal R. Clay: 1.61298713508639
Nadal R. Grass: 1.13587241578929
Nalbandian D.: -0.382553542170362
Nalbandian D. Carpet: 1.29810499889987
Nalbandian D. Clay: 0
Nalbandian D. Grass: 0
Nieminen J.: -1.69507591319015
Nieminen J. Carpet: 0
Nieminen J. Clay: 0.35503911334757
Nieminen J. Grass: 0
Nishikori K.: -0.518947279394493
Nishikori K. Clay: 0
Nishikori K. Grass: -1.01427690059201
Novak J.: -1.15019017486034
Novak J. Carpet: 0
Novak J. Clay: 0
Novak J. Grass: 0
Raonic M.: -0.699519844874133
Raonic M. Clay: -0.193902508575933
Raonic M. Grass: 0
Robredo T.: -0.979551620700338
Robredo T. Carpet: 0.107311313879213
Robredo T. Clay: 0.209926440289285
Robredo T. Grass: -0.291628913366539
Roddick A.: -0.0635296619447939
Roddick A. Carpet: 0
Roddick A. Clay: -0.315708902065761
Roddick A. Grass: 0
Safin M.: -0.717040780929182
Safin M. Carpet: 1.41376311254603
Safin M. Clay: -0.105843914921898
Safin M. Grass: 0.155522991906964
Schuettler R.: -1.40716524591561
Schuettler R. Carpet: 0
Schuettler R. Clay: 0
Schuettler R. Grass: 0.507291146881787
Seppi A.: -1.71974340177089
Seppi A. Carpet: 0
Seppi A. Clay: 0.473352685899843
Seppi A. Grass: 0
Simon G.: -0.804247233148846
Simon G. Carpet: 0
Simon G. Clay: 0
Simon G. Grass: -0.372589016806958
Soderling R.: -0.307596638746575
Soderling R. Carpet: 0
Soderling R. Clay: 0
Soderling R. Grass: 0

Stepanek R.: -1.07152917772123
Stepanek R. Carpet: 0.349888845282768
Stepanek R. Clay: 0.182009490089498
Stepanek R. Grass: 0
Tipsarevic J.: -1.20502520002535
Tipsarevic J. Carpet: 0
Tipsarevic J. Clay: 0
Tipsarevic J. Grass: 0
Tsonga J.W.: -0.0635954823958109
Tsonga J.W. Carpet: 0
Tsonga J.W. Clay: -0.240115893939751
Tsonga J.W. Grass: 0.0588387133743099
Verdasco F.: -1.03173654686279
Verdasco F. Carpet: 0
Verdasco F. Clay: 0.485033240656906
Verdasco F. Grass: 0
Wawrinka S.: -0.41500147616422
Wawrinka S. Carpet: 0
Wawrinka S. Clay: 0.196932784354177
Wawrinka S. Grass: 0
Youzhny M.: -0.809606485623527
Youzhny M. Carpet: 0
Youzhny M. Clay: -0.511397653046353
Youzhny M. Grass: 0

A.2.2 Coefficients for the Generalised Linear Model Presented in Question 9 Without Lasso Penalty

Almagro N.: -2.92233851135341
Almagro N. Yrly.: 4.83280655861322
Anderson K.: -8.18961320921312
Anderson K. Yrly.: 10.5690653025905
Baghdatis M.: -0.968060278463334
Baghdatis M. Yrly.: 1.99562430795908
Benneteau J.: -2.72417282316774
Benneteau J. Yrly.: 4.22923992865424
Berdych T.: -3.38008269965493
Berdych T. Yrly.: 6.62025124282826
Blake J.: -1.73755342744678
Blake J. Yrly.: 3.09183087718252
Canas G.: -0.423300304128438
Canas G. Yrly.: 0.443553685180928
Chardy J.: -4.28954356945572
Chardy J. Yrly.: 5.90277964851269
Chela J.I.: -1.38981192943704
Chela J.I. Yrly.: 1.23881711702901
Cilic M.: -4.63693428590789
Cilic M. Yrly.: 7.15384140633549
Clement A.: -1.4526275684109
Clement A. Yrly.: 0.806303754557621
Coria G.: -0.495143096747622
Coria G. Yrly.: 1.15513961322161
Davydenko N.: -1.69264550555782
Davydenko N. Yrly.: 3.76810068844191
Del Potro J.M.: -2.89899659124268
Del Potro J.M. Yrly.: 6.32868406561503
Djokovic N.: -3.73832074312413
Djokovic N. Yrly.: 9.04601776164292
Federer R.: -0.653231043080011
Federer R. Yrly.: 4.86271849050628
Ferrer D.: -3.07261937466967
Ferrer D. Yrly.: 6.41033964763825
Ferrero J.C.: -0.721341259926065
Ferrero J.C. Yrly.: 1.80728441495108
Fish M.: -2.94317370323638
Fish M. Yrly.: 5.47030178857159
Fognini F.: -6.30899545686742
Fognini F. Yrly.: 8.42681404596123
Gasquet R.: -2.49896342093588
Gasquet R. Yrly.: 4.95092596061013
Gaudio G.: -0.715137074569079
Gaudio G. Yrly.: -0.295145191534679
Gonzalez F.: -1.90468869198368
Gonzalez F. Yrly.: 3.95306087785912
Grosjean S.: -0.389551884383657
Grosjean S. Yrly.: -2.76578121286328
Haas T.: -1.84188321163874
Haas T. Yrly.: 4.03037019422755
Henman T.: -0.179865447053131
Henman T. Yrly.: -2.07679086338253
Hewitt L.: 0.0417914630509537
Hewitt L. Yrly.: 0.865624770264303
Isner J.: -4.78734115575707
Isner J. Yrly.: 7.19752736075089
Karlovic I.: -2.58434301098644
Karlovic I. Yrly.: 4.32260128382341
Kiefer N.: -2.11273389367643
Kiefer N. Yrly.: 3.91562555683401
Kohlschreiber P.: -4.7334168570837
Kohlschreiber P. Yrly.: 7.14509438392964
Kuerten G.: 0.928146781316959
Kuerten G. Yrly.: -7.45785912648217
Ljubicic I.: -1.75769515077959
Ljubicic I. Yrly.: 3.4709445499173
Lopez F.: -2.73822678921302
Lopez F. Yrly.: 4.55568253789011
Mathieu P.H.: -1.70096468831439
Mathieu P.H. Yrly.: 2.19759472357451
Melzer J.: -2.58814119510362
Melzer J. Yrly.: 4.23046542346794

Monaco J.: -3.53043819626896
Monaco J. Yrly.: 5.64391785709195
Monfils G.: -2.33209106431736
Monfils G. Yrly.: 4.58086780886585
Moya C.: -1.1814887544524
Moya C. Yrly.: 1.81346184546505
Murray A.: -2.78684641059224
Murray A. Yrly.: 6.74756680958478
Nadal R.: -2.08007133152817
Nadal R. Yrly.: 7.29042145374846
Nalbandian D.: -1.08058353765649
Nalbandian D. Yrly.: 2.9159612498207
Nieminen J.: -2.72663610918979
Nieminen J. Yrly.: 3.68541422247695
Nishikori K.: -7.85013638069888
Nishikori K. Yrly.: 11.2857348977423
Novak J.: -0.521528309052813
Novak J. Yrly.: -4.0419482715213
Raonic M.: -9.79119382918845
Raonic M. Yrly.: 12.9227021572313
Robredo T.: -2.57987919904284
Robredo T. Yrly.: 4.91427860361911
Roddick A.: -0.686285533968677
Roddick A. Yrly.: 2.67125567561202
Safin M.: 0.126341760981996
Safin M. Yrly.: -1.66004386904671
Schuettler R.: -1.14060747405861
Schuettler R. Yrly.: -0.470629237854678
Seppi A.: -3.51933183047933
Seppi A. Yrly.: 5.079626102426
Simon G.: -3.53123295751009
Simon G. Yrly.: 5.99426413149435
Soderling R.: -3.46187026671854
Soderling R. Yrly.: 7.15547939247513
Stepanek R.: -2.12196535353913
Stepanek R. Yrly.: 3.75492744839882
Tipsarevic J.: -3.27230408572367
Tipsarevic J. Yrly.: 5.1839676478811
Tsonga J.W.: -3.38047403940603
Tsonga J.W. Yrly.: 6.82619296466535
Verdasco F.: -2.60563189048354
Verdasco F. Yrly.: 4.71756587293659
Wawrinka S.: -4.1216421500844
Wawrinka S. Yrly.: 7.60399313925271
Youzhny M.: -2.10715637889442
Youzhny M. Yrly.: 3.71781481800663

A.2.3 Coefficients for the Generalised Linear Model Presented in Question 9 With Lasso Penalty

Almagro N.: -1.58247912819466
Almagro N. Yrly.: 0.154150934169737
Anderson K.: -5.11153600069661
Anderson K. Yrly.: 3.99869639457758
Baghdatis M.: -0.0107223065815759
Baghdatis M. Yrly.: -2.09332198689251
Benneteau J.: -1.69743278079639
Benneteau J. Yrly.: 0
Berdych T.: -2.16075122370899
Berdych T. Yrly.: 2.07569654343623
Blake J.: -0.931664864010252
Blake J. Yrly.: -0.690087122629726
Canas G.: 0.216197425085606
Canas G. Yrly.: -2.78010534741496
Chardy J.: -2.38507955484555
Chardy J. Yrly.: 0.519446079893736
Chela J.I.: -0.626599259932255
Chela J.I. Yrly.: -2.36589318142367
Cilic M.: -3.05658416293508
Cilic M. Yrly.: 2.19053374554339
Clement A.: -0.763870599623491
Clement A. Yrly.: -2.64063555905412
Coria G.: -0.140757268449357
Coria G. Yrly.: -0.88281234309302
Davydenko N.: -0.751421011162749
Davydenko N. Yrly.: -0.294720027157989
Del Potro J.M.: -1.46832407617707
Del Potro J.M. Yrly.: 1.52490721474802
Djokovic N.: -2.45972910311664
Djokovic N. Yrly.: 4.4246797217647
Federer R.: 0.330000137737601
Federer R. Yrly.: 0.626439957200027
Ferrer D.: -1.90174112358172
Ferrer D. Yrly.: 1.94199279875939
Ferrero J.C.: 0.0192392791490614
Ferrero J.C. Yrly.: -1.85271812277328
Fish M.: -1.85857360062461
Fish M. Yrly.: 1.13031865796355
Fognini F.: -4.37044774588085
Fognini F. Yrly.: 3.05782096269498
Gasquet R.: -1.2905267728239
Gasquet R. Yrly.: 0.431217542166506
Gaudio G.: -0.173208653347126
Gaudio G. Yrly.: -3.01259606051427
Gonzalez F.: -1.01071929392894
Gonzalez F. Yrly.: 0
Grosjean S.: 0.194990676658856
Grosjean S. Yrly.: -5.62953506393741
Haas T.: -0.967177177138684
Haas T. Yrly.: 0

Henman T.: 0.367628325724333
Henman T. Yrly.: -4.8082231547768
Hewitt L.: 0.857593536280035
Hewitt L. Yrly.: -3.00532286079445
Isner J.: -2.97414686098241
Isner J. Yrly.: 1.99165059551148
Karlovic I.: -1.50117718226221
Karlovic I. Yrly.: 0
Kiefer N.: -1.23870253585157
Kiefer N. Yrly.: 0
Kohlschreiber P.: -3.37822680115612
Kohlschreiber P. Yrly.: 2.42912754087969
Kuerten G.: 1.30333331824934
Kuerten G. Yrly.: -8.9895385230303
Ljubicic I.: -0.957921170842496
Ljubicic I. Yrly.: -0.293637817866766
Lopez F.: -1.67248912096855
Lopez F. Yrly.: 0.238793116955216
Mathieu P.H.: -0.832617247611083
Mathieu P.H. Yrly.: -1.69123311418886
Melzer J.: -1.56125896806805
Melzer J. Yrly.: 0
Monaco J.: -2.16226489414305
Monaco J. Yrly.: 0.924257308917948
Monfils G.: -1.07834211480288
Monfils G. Yrly.: 0
Moya C.: -0.467694045864961
Moya C. Yrly.: -1.59909069485966
Murray A.: -1.50643819419244
Murray A. Yrly.: 2.13314095078563
Nadal R.: -0.911278794862318
Nadal R. Yrly.: 2.80996043517235
Nalbandian D.: -0.209178004523754
Nalbandian D. Yrly.: -0.997572228897285
Nieminen J.: -1.81592618675353
Nieminen J. Yrly.: -0.320456357814919
Nishikori K.: -5.36984218865631
Nishikori K. Yrly.: 5.3410543648125
Novak J.: -0.0335406464567517
Novak J. Yrly.: -6.14178189371437
Raonic M.: -5.75831995805721
Raonic M. Yrly.: 5.32709371403156
Robredo T.: -1.52870526815464
Robredo T. Yrly.: 0.612741799787476
Roddick A.: 0.153335954054963
Roddick A. Yrly.: -1.22115402484427
Safin M.: 0.810013550119168
Safin M. Yrly.: -5.04287813786599
Schuettler R.: -0.411369589046558
Schuettler R. Yrly.: -3.89492079008226
Seppi A.: -2.2016515330706
Seppi A. Yrly.: 0.4376882611887
Simon G.: -2.14430436957317
Simon G. Yrly.: 1.26519157872646
Soderling R.: -2.2593491659266
Soderling R. Yrly.: 2.61497403248098
Stepanek R.: -1.1746881316132
Stepanek R. Yrly.: -0.346096987729482
Tipsarevic J.: -1.85515158236154
Tipsarevic J. Yrly.: 0.398326311608681
Tsonga J.W.: -1.95689943282729
Tsonga J.W. Yrly.: 2.04038058921418
Verdasco F.: -1.43274165228399
Verdasco F. Yrly.: 0.25195173700213
Wawrinka S.: -2.80624657163758
Wawrinka S. Yrly.: 2.94871915718484
Youzhny M.: -1.16219009517329
Youzhny M. Yrly.: -0.381006793191973

Programs

Bradley-Terry Model

Code to Split all the Data Received into Well Formatted Training and Test Data

```
Tennis <- read.csv("http://www.damtp.cam.ac.uk/user/catam/data/II
   -10-16-2019-mensResults.csv")
Tennis$Date <- as.Date(Tennis$Date,format='%d/%m/%y')
lam_seq = c(exp(seq(log(1), log(1e-10), length.out = 1000)), 0)

invlogit <- function(x) exp(x) / (1 + exp(x))
K <- function(theta) log(1 + exp(theta))

Training_Data <- Tennis[(Tennis$Date <= as.Date("2014/12/31")) & (!is.
   na(Tennis$Date)), ]
Training_length = dim(Training_Data)[1]
Test_Data <- Tennis[(Tennis$Date >= as.Date("2015/01/01")) & (!is.na(
   Tennis$Date)), ]
Test_length = dim(Test_Data)[1]

tennis_names = levels(Tennis$Winner)
Training_winners <- as.vector(Training_Data$Winner)
Training_losers <- as.vector(Training_Data$Loser)
Test_winners <- as.vector(Test_Data$Winner)
Test_losers <- as.vector(Test_Data$Loser)
Training_Y <- rep(c(1,0), length.out = Training_length)
Test_Y <- rep(c(1,0), length.out = Test_length)
```

Code to Generate and Evaluate Model 1

```
X <- matrix(data=0, nrow=Training_length, ncol = length(tennis_names)
   -1)
colnames(X) <- tennis_names[-1]

for (i in 1:Training_length) {
  winner = Training_winners[i]
  loser = Training_losers[i]
  y_factor = Training_Y[i]*2-1
  if (winner != tennis_names[1]) {
    X[i, winner] = 1 * y_factor
  }
  if (loser != tennis_names[1]) {
    X[i, loser] = -1 * y_factor
  }
}

# Make X a sparse Matrix
X <- as(X, "sparseMatrix")

TennisGLM1 <- glmnet(X, Training_Y, lambda=0, intercept=FALSE, family=
   "binomial", standardize = FALSE, thresh = 1e-14)
summary(TennisGLM1)

coefs <- as.vector(coef(TennisGLM1, s=0))
```

```
coefs <- coefs[2: length(coefs)]
names(coefs) <- colnames(X)
write.table(coefs, "question 2 coefs.txt", col.names = F, sep = " : ",
    quote = F)

LL1 = 0
for (i in 1:Training_length) {
  row_X = X[i, ]
  Xb = row_X %*% coefs
  term = Training_Y[i]*log(invlogit(Xb)) - (1-Training_Y[i])*K(Xb)
  LL1 = LL1 + term
}
LL1 = LL1*-1/Training_length
print(paste("TennisGLM1 Log Loss for Training Set:", LL1))


Test_X <- matrix(data=0, nrow=Test_length, ncol = length(tennis_names)
   -1)
colnames(Test_X) <- tennis_names[-1]

for (i in 1:Test_length) {
  winner = Test_winners[i]
  loser = Test_losers[i]
  y_factor = Test_Y[i]*2-1
  if (winner != tennis_names[1]) {
    Test_X[i, winner] = 1 * y_factor
  }
  if (loser != tennis_names[1]) {
    Test_X[i, loser] = -1 * y_factor
  }
}

LL2 = 0
for (i in 1:Test_length) {
  row_Test_X = Test_X[i, ]
  Test_Xb = row_Test_X %*% coefs
  term = Test_Y[i]*log(invlogit(Test_Xb)) - (1-Test_Y[i])*K(Test_Xb)
  LL2 = LL2 + term
}
LL2 = LL2*-1/Test_length
print(paste("TennisGLM1 Log Loss for Test Set:", LL2))

write(sprintf("Training Data Logistic Loss %.15f
Test Data Logistic Loss %.15f", LL1, LL2), file = "question 2 LogLoss.
   txt")
```

Code to Find 68% Confidence interval for the probability that Federer beats Murray

```
d <- rep(0, 59)
names(d) <- tennis_names[-1]
d["Federer R."] = 1
d["Murray A."] = -1
pred <- t(d) %*% coefs
critval <- qnorm(0.5 + 0.68/2)
```

```
# W = matrix(0, nrow = Training_length, ncol = Training_length)
W_diag = rep(0, Training_length)
for (i in 1:Training_length) {
  theta = t(X[i, ]) %*% coefs
  W_diag[i] = invlogit(theta)*(1-invlogit(theta))
}
W = diag(W_diag)
Covar_mat = solve(t(X) %*% W %*% X)
se = sqrt(as.numeric(t(d) %*% Covar_mat %*% d))
CI_prob_Bounds <- invlogit(c(pred - crital*se, pred + crital*se))
CI_string = paste0("Confidence interval the for probability that Roger
    Federer beats Andy Murray is
[",
                   CI_prob_Bounds[1],", ", CI_prob_Bounds[2], "]")
print(CI_string)

write(CI_string, file = "question 3 CI.txt")
```

Code to Generate and Evaluate Model 2

```
tennis_surfaces = levels(Tennis$Surface)
column_names = rep("",(length(tennis_names)-1)*(length(tennis_surfaces
   )))
X2 <- matrix(data=0, nrow=Training_length, ncol = length(column_names)
   )
for (i in 2:length(tennis_names)) {
  column_names[4*(i-2)+1] <- tennis_names[i]
  for (j in 2:length(tennis_surfaces))
    column_names[4*(i-2)+(j)] <- paste(tennis_names[i], tennis_
       surfaces[j-1])
}
colnames(X2) <- column_names

Training_surfaces <- as.vector(Training_Data$Surface)
Test_surfaces <- as.vector(Test_Data$Surface)

for (i in 1:Training_length) {
  winner = Training_winners[i]
  loser = Training_losers[i]
  surface = Training_surfaces[i]
  y_factor = 2*Training_Y[i]-1
  if (winner != tennis_names[1]) {
    X2[i, winner] <- 1*y_factor
    if (surface != tennis_surfaces[4]) {
      X2[i, paste(winner, surface)] <- 1*y_factor
    }
  }
  if (loser != tennis_names[1]) {
    X2[i, loser] <- -1*y_factor
    if (surface != tennis_surfaces[4]) {
      X2[i, paste(loser, surface)] <- -1*y_factor
    }
  }
```

```
}
# Delete empty columns
X2 <- X2[, colSums(X2==0) != nrow(X2)]
# Make X2 a sparse Matrix
X2 <- as(X2, "sparseMatrix")

TennisGLM2 <- glmnet(X2, Training_Y, lambda=lam_seq, intercept=FALSE,
                     family="binomial", standardize = FALSE, thresh =
                        1e-12, maxit = 1e5)
summary(TennisGLM2)


coefs2 <- as.vector(coef(TennisGLM2, s=0))
coefs2 <- coefs2[2: length(coefs2)]
names(coefs2) <- colnames(X2)

write.table(coefs2, "question 4 coefs.txt", col.names = F, sep = ": ",
    quote = F)

LL3 = 0
for (i in 1:Training_length) {
  row_X2 = X2[i, ]
  X2b = row_X2 %*% coefs2
  term = Training_Y[i]*log(invlogit(X2b)) - (1-Training_Y[i])*K(X2b)
  LL3 = LL3 + term
}
LL3 = LL3*-1/Training_length
print(paste("TennisGLM2 Log Loss for Training Set:", LL3))

Test_X2 <- matrix(data=0, nrow=Test_length, ncol = (length(tennis_
   names)-1)*(length(tennis_surfaces)))
colnames(Test_X2) <- column_names

for (i in 1:Test_length) {
  winner = Test_winners[i]
  loser = Test_losers[i]
  surface = Test_surfaces[i]
  y_factor = 2*Test_Y[i]-1
  if (winner != tennis_names[1]) {
    Test_X2[i, winner] <- 1*y_factor
    if (surface != tennis_surfaces[4]) {
      Test_X2[i, paste(winner, surface)] <- 1*y_factor
    }
  }
  if (loser != tennis_names[1]) {
    Test_X2[i, loser] <- -1*y_factor
    if (surface != tennis_surfaces[4]) {
      Test_X2[i, paste(loser, surface)] <- -1*y_factor
    }
  }
}
# Delete columns not in X2
Test_X2 <- Test_X2[, colnames(Test_X2) %in% colnames(X2)]
```

```
LL4 = 0
for (i in 1:Test_length) {
  row_Test_X2 = Test_X2[i, ]
  Test_X2b = row_Test_X2 %*% coefs2
  term = Test_Y[i]*log(invlogit(Test_X2b)) - (1-Test_Y[i])*K(Test_X2b)
  LL4 = LL4 + term
}
LL4 = LL4*-1/Test_length
print(paste("TennisGLM2 Log Loss for Test Set:", LL4))

write(sprintf("Training Data Logistic Loss %.15f
Test Data Logistic Loss %.15f", LL3, LL4), file = "question 4 LogLoss.
   txt")
```

Code to Perform a Formal Hypothesis Test on Models 1 and 2

```
test_stat = (LL1*Training_length*2-LL3*Training_length*2)
# Variance known to be 1 so use chi squared dist
p_val = 1-pchisq(test_stat, dim(X2)[2]-dim(X)[2])# length(coefs2)-
   length(coefs))
print(paste("P-value for the likelihood ratio test is:",p_val))
write(sprintf("Test Statistic: %.15f
P-value for the likelihood ratio test: %.15f", test_stat, p_val), file
    = "question 5 results.txt")
```

Regularisation

Code to Generate Model 3

```
w <- rep(c(0,1,1,1), length(tennis_names)-1)
names(w) <- column_names
w <- w[names(w) %in% colnames(X2)]

TennisLassoGLM1 <- cv.glmnet(X2, Training_Y, family="binomial", alpha
   =1, penalty.factor=w, intercept=FALSE,
                              standardize = FALSE, thresh = 1e-14,
                                 maxit = 1e5, nfolds = 10)
min_lambda = TennisLassoGLM1$lambda.min
temp_coefs = coef(TennisLassoGLM1, s=min_lambda)
coefs3 <- as.vector(temp_coefs)
names(coefs3) <- rownames(temp_coefs)
coefs3 <- coefs3[-1]
print(paste("Minimum Lambda is:", min_lambda))

write.table(coefs3, "question 7 coefs.txt", col.names = F, sep = ": ",
    quote = F)
write(paste("Lambda which minimises the mean cross-validated error is:
   ", min_lambda), file = "question 7 minLam.txt")
```

Code to Find the Number of Non-Zero Surface Terms in the Coefficients Found for Model 3

```
non_zero_surface_terms = 0
```

```
for (surface_param in names(coefs3)[!names(coefs3) %in% tennis_names])
    {
  if (coefs3[surface_param] != 0) {
    non_zero_surface_terms = non_zero_surface_terms + 1
  }
}
print(paste("The number of non zero surface terms are:", non_zero_
   surface_terms))
write(paste("The number of non zero surface terms are:", non_zero_
   surface_terms), file="question 8 nonzero.txt")
```

Code to Evaluate Model 3

```
LL5 = 0
for (i in 1:Training_length) {
  row_X2 = X2[i, ]
  X2b = row_X2 %*% coefs3
  term = Training_Y[i]*log(invlogit(X2b)) - (1-Training_Y[i])*K(X2b)
  LL5 = LL5 + term
}
LL5 = LL5*-1/Training_length
print(paste("TennisLassoGLM1 Log Loss for Training Set:", LL5))

LL6 = 0
for (i in 1:Test_length) {
  row_Test_X2 = Test_X2[i, ]
  Test_X2b = row_Test_X2 %*% coefs3
  term = Test_Y[i]*log(invlogit(Test_X2b)) - (1-Test_Y[i])*K(Test_X2b)
  LL6 = LL6 + term
}
LL6 = LL6*-1/Test_length
print(paste("TennisLassoGLM1 Log Loss for Test Set:", LL6))

write(sprintf("Training Data Logistic Loss %.15f
Test Data Logistic Loss %.15f", LL5, LL6), file = "question 8 LogLoss.
   txt")
```

Code to Generate and Evaluate Model 4

```
column_names = rep("", 2*(length(tennis_names)-1))
X3 <- matrix(data=0, nrow=Training_length, ncol = length(column_names)
   )
Test_X3 <- matrix(data=0, nrow=Test_length, ncol = length(column_names
   ))
for (i in 2:length(tennis_names)) {
  column_names[2*i-3] <- tennis_names[i]
  column_names[2*i-2] <- paste(tennis_names[i], "Yrly.")
}
colnames(X3) <- column_names
colnames(Test_X3) <- column_names

Training_Date <- Training_Data$Date
Test_Date <- Test_Data$Date
```

```r
for (i in 1:Training_length) {
  winner = Training_winners[i]
  loser = Training_losers[i]
  year = as.numeric(format(Training_Date[i],'%Y'))
  t = (year-2000)/(2014-2000)
  y_factor = 2*Training_Y[i]-1
  if (winner != tennis_names[1]) {
    X3[i, winner] <- 1*y_factor
    X3[i, paste(winner, "Yrly.")] <- t*y_factor
  }
  if (loser != tennis_names[1]) {
    X3[i, loser] <- -1*y_factor
    X3[i, paste(loser, "Yrly.")] <- -t*y_factor
  }
}

# Delete empty columns
X3 <- X3[, colSums(X3==0) != nrow(X3)]
# Make X3 into a sparse Matrix
X3 <- as(X3, "sparseMatrix")

TennisLassoGLM2 <- glmnet(X3, Training_Y, family="binomial", lambda =
   0,
                          intercept=FALSE, standardize = FALSE, thresh
                              = 1e-9, maxit = 1e5)
temp_coefs = coef(TennisLassoGLM2, s = 0)
coefs4 <- as.vector(temp_coefs)
names(coefs4) <- rownames(temp_coefs)
coefs4 <- coefs4[-1]

write.table(coefs4, "question 9 coefs 1.txt", col.names = F, sep = ":
   ", quote = F)

for (i in 1:Test_length) {
  winner = Test_winners[i]
  loser = Test_losers[i]
  year = as.numeric(format(Test_Date[i],'%Y'))
  t = (year-2000)/(2014-2000)
  y_factor = 2*Test_Y[i]-1
  if (winner != tennis_names[1]) {
    Test_X3[i, winner] <- 1*y_factor
    Test_X3[i, paste(winner, "Yrly.")] <- t*y_factor
  }
  if (loser != tennis_names[1]) {
    Test_X3[i, loser] <- -1*y_factor
    Test_X3[i, paste(loser, "Yrly.")] <- -t*y_factor
  }
}
# Delete columns not in X3
Test_X3 <- Test_X3[, colnames(Test_X3) %in% colnames(X3)]

LL7 = 0
for (i in 1:Training_length) {
```

```
  row_X3 = X3[i, ]
  X3b = row_X3 %*% coefs4
  term = Training_Y[i]*log(invlogit(X3b)) - (1-Training_Y[i])*K(X3b)
  LL7 = LL7 + term
}
LL7 = LL7*-1/Training_length
print(paste("TennisLassoGLM2 Log Loss for Training Set:", LL7))

LL8 = 0
for (i in 1:Test_length) {
  row_X3 = X3[i, ]
  X3b = row_X3 %*% coefs4
  term = Test_Y[i]*log(invlogit(X3b)) - (1-Test_Y[i])*K(X3b)
  LL8 = LL8 + term
}
LL8 = LL8*-1/Test_length
print(paste("TennisLassoGLM2 Log Loss for Test Set:", LL8))

write(sprintf("Training Data Logistic Loss %.15f
Test Data Logistic Loss %.15f", LL7, LL8), file = "question 9 LogLoss
   noWeights.txt")
```

Code to Generate and Evaluate Model 5

```
w2 <- rep(c(0,1), length(tennis_names)-1)
names(w2) <- column_names
w2 <- w2[names(w2) %in% colnames(X3)]

TennisLassoGLM3 <- cv.glmnet(X3, Training_Y, family="binomial", alpha
   =1, penalty.factor = w2,
                             intercept=FALSE, standardize = FALSE,
                                thresh = 1e-9, maxit = 1e5)

temp_coefs = coef(TennisLassoGLM3, s = "lambda.min")
coefs5 <- as.vector(temp_coefs)
names(coefs5) <- rownames(temp_coefs)
coefs5 <- coefs5[-1]

write.table(coefs5, "question 9 coefs 2.txt", col.names = F, sep = ":
   ", quote = F)

LL9 = 0
for (i in 1:Training_length) {
  row_X3 = X3[i, ]
  X3b = row_X3 %*% coefs5
  term = Training_Y[i]*log(invlogit(X3b)) - (1-Training_Y[i])*K(X3b)
  LL9 = LL9 + term
}
LL9 = LL9*-1/Training_length
print(paste("TennisLassoGLM3 Log Loss for Training Set:", LL9))

LL10 = 0
for (i in 1:Test_length) {
  row_X3 = X3[i, ]
```

```
  X3b = row_X3 %*% coefs5
  term = Test_Y[i]*log(invlogit(X3b)) - (1-Test_Y[i])*K(X3b)
  LL10 = LL10 + term
}
LL10 = LL10*-1/Test_length
print(paste("TennisLassoGLM2 Log Loss for Test Set:", LL10))

write(sprintf("Training Data Logistic Loss %.15f
Test Data Logistic Loss %.15f", LL9, LL10), file = "question 9 LogLoss
    Weights.txt")
```

Code to Generate Figure 1

```
Fed_Nad_Mat = matrix(0, nrow = 17, ncol=dim(X3)[2])
colnames(Fed_Nad_Mat) <- colnames(X3)
for (year in 2000:2016) {
  Fed_Nad_Mat[year - 1999, "Federer R."] = 1
  Fed_Nad_Mat[year - 1999, "Federer R. Yrly."] = (year-2000)/
     (2014-2000)
  Fed_Nad_Mat[year - 1999, "Nadal R."] = -1
  Fed_Nad_Mat[year - 1999, "Nadal R. Yrly."] = -(year-2000)/
     (2014-2000)
}
pdf("federer_nadal_probs.pdf")
plot(2000:2016, invlogit(Fed_Nad_Mat %*% coefs4), col = "blue", type="
   p", pch=19,
     xlab="Year", ylab="Probability Federer beats Nadal", ylim=c(0, 1)
        )
lines(2000:2016, invlogit(Fed_Nad_Mat %*% coefs4), col = "blue", lty
   =1)
par(new=T)
plot(2000:2016, invlogit(Fed_Nad_Mat %*% coefs5), col = "red", type="p
   ", pch=18,
     xlab="Year", ylab="Probability Federer beats Nadal", ylim=c(0, 1)
        )
lines(2000:2016, invlogit(Fed_Nad_Mat %*% coefs5), col = "red", lty=2)
legend("topright", legend=c("No Lasso penalty", "Lasso Penalty"), col=
   c("blue", "red"),
       lty=1:2, pch = 19:18)
dev.off()
```

Code to Compare Models 1 to 5

```
KB1 = LL1 + length(coefs)/Training_length
KB2 = LL2 + length(coefs)/Test_length
KB3 = LL3 + length(coefs2)/Training_length
KB4 = LL4 + length(coefs2)/Test_length
KB5 = LL5 + length(coefs3)/Training_length
KB6 = LL6 + length(coefs3)/Test_length
KB7 = LL7 + length(coefs4)/Training_length
KB8 = LL8 + length(coefs4)/Test_length
KB9 = LL9 + length(coefs5)/Training_length
KB10 = LL10 + length(coefs5)/Test_length
```

```
print("The following are the log-loss and approximations of the
   Kullback Liebler Divergence of the models with the true
   distribution")
print("TennisGLM1")
print(paste("KB: Training Set:", KB1, "Test Set:", KB2))
print(paste("LL: Training Set:", LL1, "Test Set:", LL2))
print("TennisGLM2")
print(paste("KB: Training Set:", KB3, "Test Set:", KB4))
print(paste("LL: Training Set:", LL3, "Test Set:", LL4))
print("TennisLassoGLM1")
print(paste("KB: Training Set:", KB5, "Test Set:", KB6))
print(paste("LL: Training Set:", LL5, "Test Set:", LL6))
print("TennisLassoGLM2")
print(paste("KB: Training Set:", KB7, "Test Set:", KB8))
print(paste("LL: Training Set:", LL7, "Test Set:", LL8))
print("TennisLassoGLM3")
print(paste("KB: Training Set:", KB9, "Test Set:", KB10))
print(paste("LL: Training Set:", LL9, "Test Set:", LL10))

write.table(c("Model 1",
              paste("KB: Training Set:", KB1),
              paste("LL: Training Set:", LL1, "Test Set:", LL2),
              "Model 2",
              paste("KB: Training Set:", KB3),
              paste("LL: Training Set:", LL3, "Test Set:", LL4),
              "Model 3",
              paste("KB: Training Set:", KB5),
              paste("LL: Training Set:", LL5, "Test Set:", LL6),
              "Model 4",
              paste("KB: Training Set:", KB7),
              paste("LL: Training Set:", LL7, "Test Set:", LL8),
              "Model 5",
              paste("KB: Training Set:", KB9),
              paste("LL: Training Set:", LL9, "Test Set:", LL10)),
                 file = "question 9 results.txt", sep = "\n",
            row.names = F, col.names = F, quote = F)
```

Can you outperform the betting market?

Code to generate the Markowitz Portfolio for Models 2 and 3

```
portfolio <- function(v, coeffs) {
  Bet_Data_2015 = Tennis[(Tennis$Date >= as.Date("2015/01/01")) &
                         (Tennis$Date <= as.Date("2015/12/31")) &
                         (!is.na(Tennis$Date)) &
                           (!is.na(Tennis$B365W)) &
                           (!is.na(Tennis$B365L)), ]
  num_bets = dim(Bet_Data_2015)[1]
  q = rep(0, 2*num_bets)
  Q = matrix(0, 2*num_bets, 2*num_bets)
  for (i in 1:num_bets) {
    match = Bet_Data_2015[i, ]
    winner = match$Winner
    loser = match$Loser
```

```
    bet_win = as.numeric(match$B365W)
    bet_loss = as.numeric(levels(match$B365L))[match$B365L]
    surface = match$Surface

    match_vec = rep(0, dim(X2)[2])
    names(match_vec) <- colnames(X2)
    if (winner %in% names(match_vec)) {
      match_vec[winner] = 1
      if (paste(winner, surface) %in% names(match_vec)) {
        match_vec[paste(winner, surface)] = 1
      }
    }
    if (loser %in% names(match_vec)) {
      match_vec[loser] = -1
      if (paste(loser, surface) %in% names(match_vec)) {
        match_vec[paste(loser, surface)] = -1
      }
    }

    mu_i = as.numeric(invlogit(match_vec %*% coeffs))
    q[2*i-1] = (bet_win - 1)*mu_i
    q[2*i] = (bet_loss - 1)*(1 - mu_i)
    A_i = matrix(c((bet_win - 1)^2                    , -(bet_win - 1)*(bet_
       loss - 1),
                    -(bet_win - 1)*(bet_loss - 1), (bet_loss - 1)^2
                                            ),2,2, byrow = TRUE)
    Q[c(2*i-1, 2*i), c(2*i-1, 2*i)] = A_i*mu_i*(1-mu_i)
  }
  A_mat = rbind(matrix(rep(1, 2*num_bets), nrow=1), diag(2*num_bets))
  b_vec = c(1000, rep(0, 2*num_bets))
  # Note Q will always be singular so will find approximate solution
     using Q + 10^-8
  sol = solve.QP(2*v*(Q)+diag(x=1e-8, 2*num_bets), q, t(A_mat), b_vec,
      meq=1)
  # Note that Solve.QP isn't great as it may still have -ve entries in
      the solution
  # To fix this we will translate all values s.t they are >= 0, then
     rescale all values accordingly
  solu = sol$solution - min(sol$solution)
  solu = solu*1000/sum(solu)
  return(solu)
}
```

Code to Evaluate the Markowitz Portfolio generated by Models 2 and 3 and Generate Figure 2

```
calc_portfolio_benefits <- function(pf) {
  Bet_Data_2015 = Tennis[(Tennis$Date >= as.Date("2015/01/01")) &
                          (Tennis$Date <= as.Date("2015/12/31")) &
                          (!is.na(Tennis$Date)) &
                          (!is.na(Tennis$B365W)) &
                          (!is.na(Tennis$B365L)), ]
  num_bets = dim(Bet_Data_2015)[1]
  total = -1000
```

```
  for (i in 1:num_bets) {
    match = Bet_Data_2015[i, ]
    bet_win = as.numeric(match$B365W)
    total = total + bet_win*pf[2*i-1]
  }
  return(total)
}

v = 10^seq(-3, 3, length.out =7)
profits1 = rep(0,length(v))
profits2 = rep(0,length(v))
for (i in 1:length(v)) {
  profits1[i] <- calc_portfolio_benefits(portfolio(v[i], coefs2))
  profits2[i] <- calc_portfolio_benefits(portfolio(v[i], coefs3))
}

pdf("markowitz_portfolio.pdf")
plot(log10(v), profits1, type = "p", col="blue", pch=19,
     xlab=TeX("$\\log_{10}\\nu$"), ylab="Total Profit",
     ylim=c(-1000, 0))
lines(log10(v), profits1, col="blue", lty=1)
par(new=TRUE)
# points(v, profits2)
plot(log10(v), profits2, type = "p", col="red", pch=18,
     xlab=TeX("$\\log_{10}\\nu$"), ylab="Total Profit",
     ylim=c(-1000, 0))
lines(log10(v), profits2, col="red", lty=2)
legend("topright", legend = c("Model 2", "Model 3"), col=c("blue", "
   red"), lty=c(1, 2), pch=c(19, 18), inset=0.1)
dev.off()
```

Code to Generate Kelly Fractions using Models 2 and 3

```
kelly_fraction <- function(match, coeffs, rho, grid, debug) {
  winner = match$Winner
  loser = match$Loser
  bet_win = as.numeric(match$B365W)
  bet_loss = as.numeric(levels(match$B365L))[match$B365L]
  surface = match$Surface
  match_vec = rep(0, dim(X2)[2])
  names(match_vec) <- colnames(X2)
  if (winner %in% names(match_vec)) {
    match_vec[winner] = 1
    if (paste(winner, surface) %in% names(match_vec)) {
      match_vec[paste(winner, surface)] = 1
    }
  }
  if (loser %in% names(match_vec)) {
    match_vec[loser] = -1
    if (paste(loser, surface) %in% names(match_vec)) {
      match_vec[paste(loser, surface)] = -1
    }
  }
  mu_i = as.numeric(invlogit(match_vec %*% coeffs))
```

```
  opt_win_loss = c(0,0)
  opt_win_loss_val = -Inf
  for (Win in seq(0, 1, length.out=grid + 1)) {
    for (Loss in seq(0, 1, length.out=grid + 1)) {
      if (Win+Loss > 1) {
        break
      } else {
        val = log((1-Loss) + Win*(bet_win-1))*(mu_i) + log((1-Win) +
           Loss*(bet_loss-1))*(1-mu_i)
        if (!is.na(val) && val != -Inf && val > opt_win_loss_val) {
          opt_win_loss = c(Win, Loss)
          opt_win_loss_val = val
        }
      }
    }
  }
  return(rho*opt_win_loss)
}
```

Code to Evaluate the Kelly Fractions generated by Models 2 and 3 and Generate Figure 3

```
eval_strat <- function(coeffs, rho, grid, bankroll){
  Bet_Data_2015 = Tennis[(Tennis$Date >= as.Date("2015/01/01")) &
                           (Tennis$Date <= as.Date("2015/12/31")) &
                           (!is.na(Tennis$Date)) &
                           (!is.na(Tennis$B365W)) &
                           (!is.na(Tennis$B365L)), ]
  Bet_Data_2015 = Bet_Data_2015[order(Bet_Data_2015$Date), ]
  num_bets = dim(Bet_Data_2015)[1]
  b = rep(0, num_bets+1)
  b[1]=bankroll
  for (i in 1:num_bets) {
    match = Bet_Data_2015[i, ]
    fractions = kelly_fraction(match, coeffs, rho, grid, FALSE)
    bet_win = as.numeric(match$B365W)
    b[i+1] = b[i]*(1 - fractions[2] + fractions[1]*(bet_win-1))
  }
  return(c(b, as.Date("2015/1/1"), Bet_Data_2015$Date))
}
strat_results = eval_strat(coefs2, 0.1, 1000, 1000)
par(new = FALSE)
pdf("question 13 bankroll.pdf")
plot(as.Date(strat_results[-(1:length(strat_results)/2)], origin = "
   1970-01-01"),
     strat_results[1:(length(strat_results)/2)], type="p",
     xlab = "Date", ylab = "Bankroll", pch=20, col="black")
lines(as.Date(strat_results[-(1:length(strat_results)/2)], origin = "
   1970-01-01"),
      strat_results[1:(length(strat_results)/2)], lty=1, col="black")
dev.off()
```