Bevacizumab for Metastatic Colorectal Cancer with Chromosomal Instability: An Cost-Effectiveness Analysis of a Novel Subtype across the COLOSSUS Partner Countries

Jonathan Briody (1) | Kathleen Bennett (1) | Lesley Tilson (2)

1. Data Science Centre, RCSI University of Medicine and Health Sciences, Dublin, Ireland
2. National Centre for Pharmacoeconomics, St James Hospital, Dublin 8, Ireland

**Co-authors:**

*Annette feels Rodrigo should be on this paper as he suggested considering the Angiopredict study. Annette said that as Matias Ebert, a gastroenterologist in Germany, is on both COLOSSUS and Angiopredict, he should be on the paper. This collaboration may also help us to get some of the cost data that we’ve been waiting on from Germany (in the end it didnt).*

*Annette feels that all the PIs on Angiopredict should be co-authors on this paper Matthias Ebert as mentioned above, Jochen Prehn and Bauke Ylstra. I would probably like to add Qiushi Chen, dependent on how helpful his thoughts on parametric survival analysis actually turn out to be.*

IF THIS BECOMES THE BEVACIZUMAB STUDY WE CAN INCOPORATE SOME OF THE BELOW:

<https://www.angiopredict.com/go/about_us>

Smeets, D., Miller, I. S., O’Connor, D. P., Das, S., Moran, B., Boeckx, B., … & Byrne, A. T. (2018). Copy number load predicts outcome of metastatic colorectal cancer patients receiving bevacizumab combination therapy. *Nature communications*, *9*(1), 1-16.

# Abstract # For your write-up on Bevacizumab, and particularly for the part indicating it’s not cost effective, you should use these 3 studies: Kristin, E., Endarti, D., Khoe, L. C., Taroeno-Hariadi, K. W., Trijayanti, C., Armansyah, A., & Sastroasmoro, S. (2021). Economic evaluation of adding bevacizumab to chemotherapy for metastatic colorectal cancer (mCRC) patients in Indonesia. Asian Pacific Journal of Cancer Prevention: APJCP, 22(6), 1921. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8418847/pdf/APJCP-22-1921.pdf> Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., El-Rayes, B. F., & Flowers, C. R. (2015). First-and second-line bevacizumab in addition to chemotherapy for metastatic colorectal cancer: a United States–based cost-effectiveness analysis. Journal of Clinical Oncology, 33(10), 1112. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4881313/pdf/zlj1112.pdf> Goldstein, D. A., Chen, Q., Ayer, T., Chan, K. K., Virik, K., Hammerman, A., … & Hall, P. S. (2017). Bevacizumab for Metastatic Colorectal Cancer: A Global Cost‐Effectiveness Analysis. The Oncologist, 22(6), 694-699. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5469597/pdf/onco12164.pdf>:

I’ll initially draft this as informed by:

Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., … & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. Clinical colorectal cancer, 13(4), 219-225. <https://www.sciencedirect.com/science/article/abs/pii/S1533002814000978>

**[TICK HERE:]**

and:

Guan, X., Li, H., Xiong, X., Peng, C., Wang, N., Ma, X., & Ma, A. (2021). Cost-effectiveness analysis of fruquintinib versus regorafenib as the third-line therapy for metastatic colorectal cancer in China. Journal of Medical Economics, 24(1), 339-344. <https://www.tandfonline.com/doi/pdf/10.1080/13696998.2021.1888743>

**[TICK HERE:]**

and:

Hamdy Elsisi, G., Nada, Y., Rashad, N., & Carapinha, J. (2019). Cost-effectiveness of sorafenib versus best supportive care in advanced hepatocellular carcinoma in Egypt. Journal of Medical Economics, 22(2), 163-168. <https://www-tandfonline-com.proxy.library.rcsi.ie/doi/pdf/10.1080/13696998.2018.1552432>

**[TICK HERE:]**

and here:

Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Ramalingam, S. S., … & Flowers, C. R. (2015). Necitumumab in metastatic squamous cell lung cancer: establishing a value-based cost. JAMA oncology, 1(9), 1293-1300.

Paper saved here: <file:///C:/Users/Jonathan/Dropbox/PC/Downloads/13%20Goldstein.pdf> and supplementary material saved here: <file:///C:/Users/Jonathan/Dropbox/PC/Downloads/Neci%20supplemental%20material%20071815.pdf>

**[TICK HERE:]**

and not forgetting I have some relevant stuff in Grammarly

**[TICK HERE:]**

Then I’ll have a look at the other open journal articles in my browser windows

**[TICK HERE:]**

**Background:** Colorectal cancer is the third most common cancer in Europe. Bevacizumab, a monoclonal antibody that functions as an angiogenesis inhibitor, was approved for the first-line treatment of metastatic colorectal cancer (mCRC) in 2004. However, the addition of Bevacizumab to first-line chemotherapy in mCRC consistently fails to be cost-effective due to moderate/modest response in patients. Recently, copy number load has identified a *sub*type of patients with improved outcomes following Bevacizumab combination therapy. The objective of this study was to compare the cost-effectiveness of adding Bevacizumab to first-line treatment chemotherapy, in comparison to chemotherapy alone, for such patients.

**Materials and Methods:** We developed a three-state Markov model with fortnightly cycles to estimate the incremental cost-effectiveness ratio (ICER) of Bevacizumab (Avastin) plus FOLFOX (5-FU, leucovorin, and oxaliplatin) versus FOLFOX alone. Progression risks and cause-specific mortality were informed by progression-free survival and overall-survival data from the ANGIOPREDICT study on this newly identified subtype of patient with improved responsiveness to Bevacizumab. Probabilities of adverse events were taken from the literature on Bevacizumab combination therapy. Costs for treatment administration and management of adverse events were derived from surveys of consortium hospitals in the participating countries. Utilities applied to calculate the quality-adjusted life years (QALYs) were obtained from a review of the literature. The perspective was that of a healthcare payer in the countries of the COLOSSUS translational study (Germany, Ireland and Spain), where the cost-effectiveness build was built. To test the robustness of results, parameter values were varied individually to assess the model’s sensitivity to individual parameters, and simultaneously, sampling multiple sets of parameter values from a priori–defined probability distributions to represent uncertainty in model parameters.

**Results:** In Ireland, Germany and Spain, the addition of Bevacizumab to therapy provided X QALYS at a cost of €X,000, X QALYS at a cost of €X,000, and X QALYS at a cost of €X,000 respectively, compared with standard of care (FOLFOX without Bevacizumab). Deterministic results indicate that the highest ICER was in Ireland (€,000/QALY), while the lowest was in Spain (€,000/QALY). In Germany, the ICER was €,000 per QALY. Compared to standard of care, the addition of Bevacizumab provided an ICER well above the threshold used to define cost-effectiveness from a European healthcare perspective in every country studied. All univariate and multivariate sensitivity analyses demonstrated that the initial results were generally robust. Cost-effectiveness acceptability curves from the probabilistic sensitivity analysis showed that adding Bevacizumab to therapy was only cost-effective when willingness to pay thresholds were multiple times larger than currently accepted cut-offs.

*Likewise, in the probabilistic sensitivity analysis, Bevacizumab was never identified as being cost-effective in any country at local willingness to pay thresholds.*

**Conclusion:** When compared to standard of care, predictive biomarkers with targeted therapy is cost effective for patients with MSS RAS mt mCRC in Germany, Ireland and Spain at a €X,000 per QALY threshold. However, because of the exploratory nature of this analysis, the cost effectiveness profile of these biomarkers should be evaluated in further comparative studies. Even for recently identified patients with improved outcomes following Bevacizumab combination therapy, adding Bevacizumab to first-line chemotherapy for mCRC invariably is not cost-effective. This result remains unchanged across variability in pricing, healthcare costs, willingness to pay thresholds and multiple international payers.

**Keywords:** Colorectal neoplasms, cost effectiveness analysis, Quality-adjusted life years, Markov model, mCRC,

JEL CLASSIFICATION CODES

I19, I1, I, I11, I1, I,

*Possible Journal Homes:*

The Oncologist: Published a study on this in 2017:

Goldstein, D. A., Chen, Q., Ayer, T., Chan, K. K., Virik, K., Hammerman, A., ... & Hall, P. S. (2017). Bevacizumab for Metastatic Colorectal Cancer: A Global Cost‐Effectiveness Analysis. *The Oncologist*, *22*(6), 694-699.

Journal of Medical Economics <https://www.tandfonline.com/doi/pdf/10.1080/13696998.2021.1888743>

Clinical Colorectal Cancer <https://www-sciencedirect-com.proxy.library.rcsi.ie/science/article/pii/S1533002814000978>

Value in Health: FOCUSED ON CEA, ETC.,

Studies in Health Technology and Informatics

“New Journal: Oxford Open Economics” I have an email on this in my gmail, it’s a new Journal and thus may be interested in this work.

We can increase our publication numbers by publishing a short summary of our study in: PharmacoEconomics & Outcomes News - per the description “Stay up to date with the exciting world of health economics. Solve the problem of reading and monitoring the vast volumes of literature necessary to stay informed on the latest issues in pharmacoeconomics and outcomes research by reading our concise news summaries, compiled in an easy-to-read format.” <https://www.springer.com/journal/40274>

<https://www.clinical-colorectal-cancer.com/>

**Correspondence:**

Jonathan Briody, PhD, RCSI University of Medicine and Health Sciences, Beaux Lane House, Lower Mercer St, Dublin 2, Ireland. Email: [jonathanbriody@rcsi.ie](mailto:jonathanbriody@rcsi.ie)

**Funding Information:**

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 754923. The material presented and views expressed here are the responsibility of the author(s) only. The EU Commission takes no responsibility for any use made of the information set out.

# Introduction

With almost 2 million new cases diagnosed annually, colorectal cancer is the third most commonly diagnosed cancer in men, the second most commonly diagnosed in women (Vabi, Gibbs, and Parker 2021), and the second leading cause of cancer deaths worldwide (Nwaokorie and Fey 2021). Following advances in primary and adjuvant treatments, survival time continues to improve in colorectal cancer, with a 5-year survival rate of 64%, however, this is not reflected in metastatic colorectal cancer (mCRC), which has a dramatically lower 5-year survival rate of just 12% (Siegel, Miller, and Jemal 2019).

Cancerous cells may be addressed by targeted therapy which directly inhibits cell proliferation, differentiation, and migration. Similarly, the micro-environment of the tumor, including it’s local blood vessels and immune cells, may be modified in order to disrupt growth and to improve immune surveillance and response (Tiwari et al. 2018). Numerous targeted therapies for colorectal cancer have successfully come to market in the last two decades. As the first clinically applied angiogenesis inhibitor in any disease, Bevacizumab, *a monoclonal antibody that functions as an angiogenesis inhibitor,* may be the most successful of these (Shih and Lindley 2006).

Although widely used as an addition to the chemotherapeutic regimen in the first-line treatment of metastatic colorectal cancer (mCRC), Bevacizumab is typically not considered cost-effective [Kristin et al. (2021)](Goldstein, Chen, Ayer, Howard, Lipscomb, El-Rayes, et al. 2015b)(Franken et al. 2017). This may correspond to a limited overall benefit of Bevacizumab in the clinic [@potti2010] (Goldstein, Chen, Ayer, Howard, Lipscomb, El-Rayes, et al. 2015a). However, there is emerging evidence that for a subtype of mCRC patients, the effect of Bevacizumab is beyond what has previously been seen in mCRC. The ANGIOPREDICT study investigated responsiveness to Bevacizumab per degree of chromosomal instability, identifying a subgroup of mCRC patients with improved outcomes following addition to chemotherapy treatment [@betge2016].

ANGIOPREDICT patients demonstrated significantly improved PFS (HR = 0.68; P = 0.00249; 95% CI 0.53–0.87) and OS (HR = 0.65; P = 0.00227; 95% CI 0.49−0.86) after adding Bevacizumab to treatment compared to chemotherapy alone. Importantly, such improvements were also therapeutically relevant, with a more prolonged median survival of X days compared to chemotherapy alone [@smeets2018]. We hypothesised that such improvements in effectiveness might translate into improvements in the cost-effectiveness of Bevacizumab treatment for mCRC patients with chromosomal instability, providing greater support for clinical use [@jamison2006]. Consequently, the objective of the present study was to conduct a health economic analysis of treatment in this emergent group.

# Materials and Methods

an

#### Model Structure

Limited data exist regarding these novel therapies, with no research comparing the cost-effectiveness of these targeted therapies with standard of care, thus, we developed a general decision-analytic framework applied to mCRC to model the cost-effectiveness of biomarkers for targeted therapies in mCRC. We constructed a Markov model using a hypothetical cohort of patients with mCRC to conduct an exploratory cost-effectiveness analysis of targeted treatment based on new diagnostic tests vs standard of care in MSS RAS mt mCRC patients.

We constructed Markov models to calculate the lifetime costs and outcomes of each treatment strategy. As per Figure 1, in this model we established the following health states: (1) first line treatment prior to progression; (2) second line treatment following progression; (3) death. We applied published estimates of costs associated with each health state in the countries studied [CITE]. All cost estimates were in 2022 Euros [maybe discuss how you converted to this? HIQA has something on this I think]. To apply country-specific state utility values we…DETAIL HERE THE APPROACH WE TOOK AS DISCUSSED WITH KATHLEEN [CITE]. The model was programmed in R 4.1.2 using RStudio 2021.09.2+382 and was informed by the state-transition model framework developed by the Decision Analysis in R for Technologies in Health (DARTH) workgroup (Alarid-Escudero et al. 2021) (Jalal et al. 2017) (Krijkamp et al. 2018) (Krijkamp et al. 2020). The Markov model simulated the health state transitions following a diagnosis of metastatic cancer. A cycle length of 1 month was chosen, per the time between administration of chemotherapy doses. A 5-year time horizon was also chosen, this was appropriate to estimate life expectancy in this patient population [CITE]. For standard of care, we considered 2 lines of treatment, first line (FOLFOX) and second line (FOLFIRI). For the comparator we also considered 2 lines of treatment (NEW THERAPY), then (FOLFIRI). In both instances we made use of treatment-specific utility values. At the end of each cycle, patients remained in their current health state, or transitioned to a more progressive state. To reflect that transitions between states may occur at any point within each cycle, we applied a half-cycle correction to calculating life-years (LYs) {I AM DEFINITELY NOT GOING TO DO THIS, I JUST PUT IT IN HERE AS SOMETHING TO THINK ABOUT, CYCLES ARE ONLY A MONTH SO A CORRECTION OF 2 WEEKS IS PRETTY STUPID}. Within cycle correction here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\R\Functions.R

[A three-state Markov model (Figure 1) with monthly cycle could also have been used to represent the progression of mCRC, including PFS, progressive disease (PD), and death. All patients entered the model either receiving FOLFOX therapy or NOVEL therapy. After treatment discontinuation due to progession, best supportive care (BSC) was modeled. \*\*–> So, here even doing that they still got a treatment that needs to be modelled in PFS].\*\*

Guan, X., Li, H., Xiong, X., Peng, C., Wang, N., Ma, X., & Ma, A. (2021). Cost-effectiveness analysis of fruquintinib versus regorafenib as the third-line therapy for metastatic colorectal cancer in China. *Journal of Medical Economics*, *24*(1), 339-344.

The intention-to-treat (ITT) population we modeled in this study is MSS RAS mt mCRC patients in Germany, Ireland and Spain, from a healthcare payer perspective in these countries. Subsequently the analysis only included direct medical costs. Effectiveness is reflected in life-years (LYs) and quality adjusted life years (QALYS). Per the countries studied, a discount rate of 4% per annum was applied to cost input and health output parameters [citation here]. The primary outcome of the model was the total direct costs, LYs, QALYS, and the incremental cost-effectiveness ratio (ICER).

#### Patients and Treatment Regimens

In the absence of a randomised controlled trial making direct comparisons between standard of care and the proposed novel therapy, it was necessary to perform an indirect comparison using the data from two published studies [CITE]. We made use of published results from one study to estimate the clinical course for patients treated under standard of care and another to estimate outcomes for patients under novel therapy (Goldstein et al. 2014a). The standard of care arm was based on a study be ..et al., {FILL IN THE BLANKS HERE ON THE STUDY YOU BASED THIS ON} who compared…[cite]. The novel therapy arm was based on a X, Y, Z, study by ….et al., While these 2 studies did not have identical target populations…

The current front-line standard of care for mCRC is FOLFOX, given with Bevacizumab. Patients who progress on first-line therapy typically receive second-line treatment with 5-FU, leucovorin, and irinotecan (FOLFIRI) [@tournigand2004].

smeet

The PK arm was based on a phase II nonrandomized study by Capitain et al., who compared PK FOLFOX with BSA FOLFOX.9 The BSA arm was based on a phase III randomized study by Tournigand et al.14 Although the 2 studies did not have identical target populations, the survival outcomes of the baseline BSA arm of the 2 studies are comparable (Table 1).9,14,15 Moreover, the median PFS and OS were also similar to the FOLFOX arm in a similar study.15

Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., … & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. *Clinical colorectal cancer*, *13*(4), 219-225.

Given that there was no single RCT with all treatment arms, it was necessary to perform an indirect comparison using the data from the FRESCO trial and the CONCUR trial6, 11. The baseline treatment for the indirect comparison was BSC taken from the FRESCO trial. The clinical effectiveness in patients in the BSC arm was also available from the CONCUR trial. This, however, was not considered appropriate for the baseline treatment because the survival curves reported by the CONCUR trial were distorted in the paper and were not suitable for recreating individual patient data. However, the hazard ratios (HRs) calculated by original data would not be affected by the distorted figures. The implicit assumption underlying the indirect comparison was that the baseline patient characteristics in the two studies were reasonably similar, which was shown to be true (Appendix Table 1).

[I THINK WE CAN TAKE THEIR APPROACH IN OUR MODEL, DO DIGITISING AND PARAMETRIC SURVIVAL ANALYSIS FOR STANDARD OF CARE, THEN TAKE A HAZARD RATIO FROM ANNETTE AND THE COLOSSUS STUDY FOR THE NEW INTERVENTION, THERE’S A LITTLE MORE ON HR’S BELOW and the following R code applies hazard ratios after doing parametric survival analysis, similar to what I plan to do: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\ISPOR WEBINAR Health Economic Modelling in R\ISPOR\_webinar\_R-master\ISPOR\_webinar\_R-master\oncologySemiMarkov\_illustration.R].

The FRESCO trial was used to model the probabilities of remaining in the PFS state or experiencing disease progression in the fruquintinib arm. This research recreating individual patients data (IPD) from Kaplan–Meier (KM) survival curves in FRESCO by the method published by Guyot et al.12. To predict the events of death and progression beyond the observation period, several parametric models were fit in different distributions, including exponential, Weibull, Gompertz, generalized gamma log-normal and log-logistic. After comparing the goodness-of-fit of different parametric models by the Akaike information criterion (AIC) and the Bayesian information criterion (BIC) (Appendix Table 2), assessing the conformities of parametric models to the original KM survival curves in the FRESCO trial and considering about proportional hazard model used for the indirect comparison, the Weibull distribution model was chosen for the base case analysis (Appendix Figure 1 and Appendix Figure 2).

We used the PFS curves to estimate the transition probability from PFS state to PFS state and the transition probability from PFS state to another states (PD and death) and used the OS curve to estimate the overall mortality. The model assumed the transition probability from PFS state to death was estimated by the background mortality obtained from Chinese government data, which based on actuarial estimates of mortality among the general population in China. According to the background mortality and the transition probability from PFS state to another states (PD and death), we calculated the transition probability from PFS state and PD. Moreover, the transition probability from PD state to PD state could calculated by overall mortality and background mortality.

We also calculated the transition probabilities in the BSC arm by a Weibull-distribution hazard function using the survival data from the FRESCO trial. The model assumed that the survival curves of BSC both in FRESCO and CONCUR would be identical if the patients in BSC arm was identical, and the HR of regorafenib versus BSC would not change even if the patients receiving BSC in CONCUR became exactly same as the patients receiving BSC in FRESCO. Hence,we adjusted the transition probabilities of the regorafenib arm using the transition probabilities of the BSC arm modeled from the FRESCO trial and the HR of regorafenib versus BSC taken from the CONCUR trial.

Guan, X., Li, H., Xiong, X., Peng, C., Wang, N., Ma, X., & Ma, A. (2021). Cost-effectiveness analysis of fruquintinib versus regorafenib as the third-line therapy for metastatic colorectal cancer in China. *Journal of Medical Economics*, *24*(1), 339-344.

There is a discussion on indirect comparisons and how best to do these in this short piece: Martin-Broto, J. (2015). Indirect comparisons in cost-effectiveness analysis: are we being naïve?. Clinical and Translational Oncology, 17(1), 85-86. <https://sci-hub.se/10.1007/s12094-014-1256-9>

This also talks about indirect comparisons Hollmann, S., Alloul, K., Attard, C., & Kavan, P. (2014). An Indirect Treatment Comparison and Cost-Effectiveness Analysis Comparing Folfirinox with Nab-Paclitaxel Plus Gemcitabine for First-Line Treatment for Patients with Metastatic Pancreatic Cancer. Annals of Oncology, 25, ii11. <https://sci-hub.se/10.1093/annonc/mdu164.18>

This applies an indirect comparison in more detail: Tamoschus, D., Draexler, K., Chang, J., Ngai, C., Madin-Warburton, M., & Pitcher, A. (2017). Cost-effectiveness analysis of regorafenib for gastrointestinal stromal tumour (GIST) in Germany. Clinical Drug Investigation, 37(6), 525-533. <https://sci-hub.se/10.1007/s40261-017-0514-3>

I’m not even sure if I would classify what we are doing as an indirect comparison, as we don’t have data for the novel therapy at all (that I know of), but I’ll read the above articles and then make a call on that. Likely I’ll make the call to follow the approach of: Guan, X., Li, H., Xiong, X., Peng, C., Wang, N., Ma, X., & Ma, A. (2021). Cost-effectiveness analysis of fruquintinib versus regorafenib as the third-line therapy for metastatic colorectal cancer in China. *Journal of Medical Economics*, *24*(1), 339-344. and use a HR, but I’ll see.

The following also shows how to do the above: Purmonen, T., Martikainen, J. A., Soini, E. J., Kataja, V., Vuorinen, R. L., & Kellokumpu-Lehtinen, P. L. (2008). Economic evaluation of sunitinib malate in second-line treatment of metastatic renal cell carcinoma in Finland. Clinical therapeutics, 30(2), 382-392. - <https://sci-hub.se/10.1016/j.clinthera.2008.02.013>

***Or we could try:***

The transition probabilities were derived from the median time to radiologic progression and median OS obtained from previously published studies18,20 following recommended methods21. - Per: Hamdy Elsisi, G., Nada, Y., Rashad, N., & Carapinha, J. (2019). Cost-effectiveness of sorafenib versus best supportive care in advanced hepatocellular carcinoma in Egypt. Journal of Medical Economics, 22(2), 163-168. <https://www-tandfonline-com.proxy.library.rcsi.ie/doi/pdf/10.1080/13696998.2018.1552432>

I think the below also takes transition probabilities just from the literature, although I’d have to read it and see:

Pierse, T., O’Neill, S., Dinneen, S. F., & O’Neill, C. (2021). A simulation study of the economic and health impact of a diabetes prevention programme in Ireland. *Diabetic Medicine*, *38*(6), e14540.

#### Transition Probabilities

##### Life Tables:

Following significant review of the literature on CEA of mCRC it’s pretty clear that it’s extremely uncommon for published studies to make use of life tables, most likely because of the short life-span of mCRC patients. However, if I did need to make use of Life Tables for this model, I would utilise the following:

WRITING:

Non mCRC death was captured as a competing risk in the model. We use life tables in each country studied to estimate the background mortality, these data were derived from [CITE WHERE YOU GOT THE LIFE TABLES] for GIVE THE YEAR. The starting age of the cohort was X years, which is the median age of incident mCRC diagnosis, as reported by [SOME TRUST WORTHY CANCER REGISTRATION BODY].

CODE:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A1\_Advanced Markov Modelling\A1.3 Practical exercise R

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_24\4\_cSTM - time-dependent models\_material

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_25\3\_cSTM - history dependence\_material

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.3 Practical exercise R

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A3\_Presenting Simulation Results\A3.3 Practical exercise R

# Parametric Survival Analysis:

Some more R code on parametric survival analysis here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Applied Cost Effectiveness Modeling with R\rcea-rscripts\05-psm

This study: Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Ramalingam, S. S., … & Flowers, C. R. (2015). Necitumumab in metastatic squamous cell lung cancer: establishing a value-based cost. JAMA oncology, 1(9), 1293-1300. saved here: C:/Users/Jonathan/Dropbox/PC/Downloads/coi150070.pdf and particulary it’s supplementary material, saved here: <file:///C:/Users/Jonathan/Dropbox/PC/Downloads/supplementary%20materials.pdf> explains how to use PFS and OS curves to generate transition probabilities from the first state into the second and from the first state into death, so, it tells you how to create both by taking the PFS probabilities and OS probabilities from eachother to get the progressing probability and using the OS probability to get the death probability, it also talks about AIC, BIC, etc.,

Something to consider in this is the following point:

“Model Validation We performed internal model validations demonstrating that the overall survival curves generated by the Markov model simulation closely approximated those presented in the SQUIRE trial14 and the fitted survivalmodels (see eFigure 3 and eFigure 4 in the Supplement).”

And I talked to **Qiushi Chen** about this in detail in email (I also described to Kathleen in email), so look to that conversation on how he approached this to make sure I am not making any mistakes.

This talks about AIC as well: Briggs, A., Sculpher, M., Dawson, J., Fitzpatrick, R., Murray, D., & Malchau, H. (2003). Modelling the cost-effectiveness of primary hip replacement: how cost-effective is the Spectron compared to the Charnley prosthesis. CHE Technical Paper Series, 28. <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Training%20Resources/Decision%20Modelling%20-%20Advanced%20Course/A2_Making%20Models%20Probabilistic/A2.1.2%20Distributions%20for%20parameters/gammadist/interquartile%20range.pdf>

A GREAT EXPLANATION OF PARAMETRIC SURVIVAL ANALYSIS, MATHEMATICAL NOTATION AND CODE CAN BE FOUND HERE: <https://rpubs.com/mbounthavong/survival_analysis_in_r> ALSO SAVED HERE: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\rpubs-com-mbounthavong-survival\_analysis\_in\_r

Info here too: <https://devinincerti.com/2019/06/18/parametric_survival.html>

Survival Curves in R: <https://www.emilyzabor.com/tutorials/survival_analysis_in_r_tutorial.html> <https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_survival/BS704_Survival_print.html>

Progression and mortality probabilities were derived from the PFS and OS curves of the X study [CITE]. Data points were extracted from survival plots using Engauge Digitizer software, version 12.2, and these data points were used to fit parametric survival models (Mitchell 2022).

To produce estimates of progression and mortality outside of the follow-up time in X study several parametric models were fit to a number of distributions, including exponential, Weibull, Gompertz, generalized gamma log-normal and log-logistic. Statistical analyses demonstrated a good fit was provided for both curves by Weibull models, according to the Akaike information criterion (AIC), the Bayesian information criterion (BIC), Schwarz Bayesian criterion and visual inspection of the relative conformity of parametric models to the original Kaplan-Meier survival curves (Table A1 in the appendix, Figure A1 in the appendix). Weibull models have been demonstrated to be more suitable for modeling events occurring in mCRC (Goldstein et al. 2014b), and support the application of proportional hazard models (Guan et al. 2021) such that the Weibull distribution model was chosen for the base case analysis. Therefore, we estimated the transition probabilities for each cycle based on the fitted Weibull survival models.

The cycle length studied was 2 weeks, corresponding to the time between chemotherapy administrations. Because PFS curves capture patients under progression and death, we estimate the progression risks, that is the transition probability from first to second line treatment, by subtracting the mortality risks $r\_{i}$ from these transition probabilities $p\_{i}$. Thus, the probability of progression is = max($p\_{i}$ - $r\_{i}$, 0), *i* = 1,2,\cdots, per (Goldstein, Chen, Ayer, Howard, Lipscomb, Ramalingam, et al. 2015).

Transition from first to second line treatment in each cycle was calculated based on the (Wang et al. 2020) [USE THIS PAPER TO DESCRIBE THE PARAMETRIC SURVIVAL ANALYSIS] formulation: 1−exp($*λ*[t−1]^{γ}$ − $*λ*t^{γ}$), where the current stage in the model is t, and *λ* and *γ* are the estimated scale and shape parameters respectively. The transition probability of second-line treatment into death is as informed by (Wen et al. 2016).

**3 TIMES GDP THRESHOLD:**

We used the World Health Organization’s Choosing Interventions that are Cost–Effective (WHO-CHOICE) suggestion that “interventions for a given country or region that cost less than three times average per capita income per disability-adjusted life-year (DALY) averted are considered costeffective; and those that exceed this level are considered not cost-effective” (estimation from CAPMAS)29. The three times GDP per capita in Egypt in 2017 was EGP106,000, which corresponds to an incremental cost-effectiveness ratio (ICER) threshold of USD 41,372 (2017)30. Hamdy Elsisi, G., Nada, Y., Rashad, N., & Carapinha, J. (2019). Cost-effectiveness of sorafenib versus best supportive care in advanced hepatocellular carcinoma in Egypt. Journal of Medical Economics, 22(2), 163-168. <https://www-tandfonline-com.proxy.library.rcsi.ie/doi/pdf/10.1080/13696998.2018.1552432>

“The WTP threshold was set as three times the GDP per capita in 2017, which was $10,800/QALY in Indonesia.”

Kristin, E., Endarti, D., Khoe, L. C., Taroeno-Hariadi, K. W., Trijayanti, C., Armansyah, A., & Sastroasmoro, S. (2021). Economic evaluation of adding bevacizumab to chemotherapy for metastatic colorectal cancer (mCRC) patients in Indonesia. *Asian Pacific Journal of Cancer Prevention: APJCP*, *22*(6), 1921.

<file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Evidence%20Synthesis/Economic%20Models/Kristin%20et%20al_2021_Economic%20Evaluation%20of%20Adding%20Bevacizumab%20to%20Chemotherapy%20for%20Metastatic.pdf>

three times the GDP per capita of China Cost-effectiveness analysis of fruquintinib versus regorafenib as the third-line therapy for metastatic colorectal cancer in China tandfonline.com/doi/pdf/10.1080/13696998.2021.1888743

Maybe for \*\*Germany\*\*, where they don’t appear to have a cost-effectiveness threshold, I can use the GDP per capita approach I’ve read about (somewhere) or some range of this. Rough description of this here: Marseille, E., Larson, B., Kazi, D. S., Kahn, J. G., & Rosen, S. (2014). Thresholds for the cost–effectiveness of interventions: alternative approaches. Bulletin of the World Health Organization, 93, 118-124. <https://www.who.int/bulletin/volumes/93/2/14-138206/en/> Health Opportunity cost may also factor into this: Ochalek, J., Lomas, J., & Claxton, K. (2018). Estimating health opportunity costs in low-income and middle-income countries: a novel approach and evidence from cross-country data. BMJ Global Health, 3(6). <https://gh.bmj.com/content/3/6/e000964>

This study also didnt have thresholds, much like Germany doesnt: Franken, M. D., Van Rooijen, E. M., May, A. M., Koffijberg, H., van Tinteren, H., Mol, L., … & van Oijen, M. G. H. (2017). Cost-effectiveness of capecitabine and bevacizumab maintenance treatment after first-line induction treatment in metastatic colorectal cancer. European Journal of Cancer, 75, 204-212. <https://sci-hub.ru/10.1016/j.ejca.2017.01.019>

# 00 Set up general state of play

rm(list = ls())   
# clear memory (removes all the variables from the work space)  
options(scipen=999)   
# turns scientific notation off  
# options(scipen=0) turns it back on, per: https://stackoverflow.com/questions/5352099/how-can-i-disable-scientific-notation

# 01 Load all packages

# Joshua's Package Manager:  
  
# Package names  
  
packages <- c("pacman", "flexsurv", "MASS", "dplyr", "devtools", "scales", "ellipse", "ggplot2", "lazyeval", "igraph", "ggraph", "reshape2", "knitr", "stringr", "diagram", "survival", "lubridate", "ggsurvfit", "gtsummary", "tidycmprsk", "magrittr", "dplyr", "diffr")  
  
# Install packages not yet installed  
  
installed\_packages <- packages %in% rownames(installed.packages())  
  
if (any(installed\_packages == FALSE)) {  
  
 install.packages(packages[!installed\_packages])  
  
}  
  
# Packages loading  
  
invisible(lapply(packages, library, character.only = TRUE))  
  
p\_load\_gh("DARTH-git/dampack", "DARTH-git/darthtools")

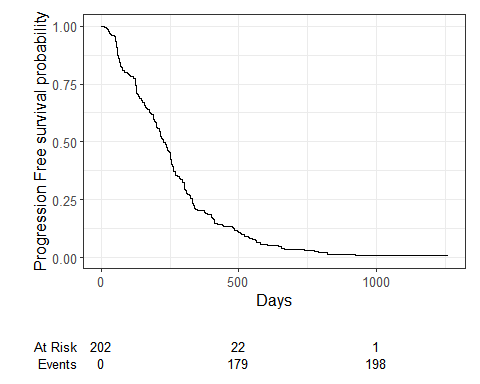
# Check whether the required packages are installed and, if not, install the missing packages  
# # - the 'pacman' package is used to conveniently install other packages  
# if (!require("pacman")) install.packages("pacman"); library(pacman)  
# p\_load("flexsurv", "MASS", "dplyr", "devtools", "scales", "ellipse", "ggplot2", "lazyeval", "igraph", "ggraph", "reshape2", "knitr", "stringr", "diagram")   
# p\_load\_gh("DARTH-git/dampack", "DARTH-git/darthtools")

# 02 Individual Data for Parametric Code

# Load the individual patient data for the time-to-progression (TTP) that you recovered by digitising published survival curves  
  
# df\_TTP <- read.csv(file='PFS.csv', header=TRUE)  
# save.image("C:/Users/Jonathan/OneDrive - Royal College of Surgeons in Ireland/COLOSSUS/R Code/GitHub/COLOSSUS\_Model/df\_TTP.RData")  
# df\_TTD <- read.csv(file='OS.csv', header=TRUE)  
# save.image("C:/Users/Jonathan/OneDrive - Royal College of Surgeons in Ireland/COLOSSUS/R Code/GitHub/COLOSSUS\_Model/df\_TTD.RData")  
  
load(file = "df\_TTP.RData")  
# Load the individual patient data for the time-to-death (TTD) that you recovered by digitising published survival curves  
load(file = "df\_TTD.RData")  
# At the moment I don't have time-to-death (TTD) data, and creating a fake dataset by duplicating time-to-progression (TTP) won't work as R will recognise the duplication and refuse to load the data, but when I DO have time-to-death (TTD) data I'll load it in here.  
  
# Here's how to remove the BVZ rows from the excel files we have: https://techcommunity.microsoft.com/t5/excel/deleting-rows-that-contain-specific-content/m-p/2084473  
  
# At the moment this is just nonsense data and needs to be replaced when I have figured out the issues described at the end of this section.  
  
# When I do have data, the data needs to be set up to include a column for the time (in the example this is time in years) and a status indicator whether the time corresponds to an event, i.e. progression (status = 1), or to the last time of follow up, i.e. censoring (status = 0),  
  
# "What is Censoring? Censoring in a study is when there is incomplete information about a study participant, observation or value of a measurement. In clinical trials, it's when the event doesn't happen while the subject is being monitored or because they drop out of the trial." - https://www.statisticshowto.com/censoring/#:~:text=What%20is%20Censoring%3F,drop%20out%20of%20the%20trial.  
  
# https://en.wikipedia.org/wiki/Survival\_analysis#:~:text=or%20q%20%3D%200.99.-,Censoring,is%20common%20in%20survival%20analysis.  
  
# At the bottom of the parametric survival code I think about how time from individual patient data may be changed to match the time of our cycles.  
  
# I think the digitiser will give me the time in the context of the survival curves I am digitising, i.e., time in weeks, or time in months or time in years.  
  
# Then I will have to set-up my time accordingly in the R code so that my cycle length is at the same level as the individual patient data.  
  
# That is, in Koen's example:  
  
# C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\ISPOR WEBINAR Health Economic Modelling in R\ISPOR\_webinar\_R-master  
  
# the data includes a column for the time (in years)  
# t\_cycle <- 1/4 # cycle length of 3 months (in years) # n\_cycle <- 60 # number of cycles (total runtime 15 years)  
   
  
# So I would have my colum for time, in the [TIME] the graph I was digitising used.  
# Then I would create my cycle length of X weeks (in [TIME] the graph I was digitising used)  
# Then I would have my number of cycles that would add up to give me a total runtime of how long I want to run the model for.  
# So, above Koen wanted to run the model for 15 years, but his cycles were in 3 months, or each cycle was a quarter of a year, so 60 quarters, or 60 3 month cycles, is 15 years.

# 03 Parametric Survival Analysis Model Plan

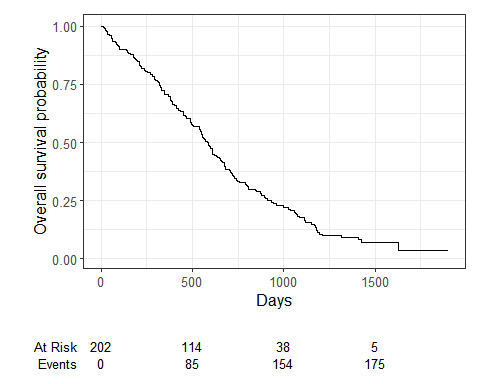
# We will implement a semi-Markov model with time-dependent transition probabilities, via a parametric survival model fitted to some individual patient data for the time-to-progression (TTP) and time-to-death (TTD) for standard of care.  
  
# A hazard ratio for the new intervention therapy vs. the standard of care will then be applied to obtain transition probabilities for the new experimental strategy.  
  
# Let's first review the survival curves and see if they match the study:  
  
# Simple survival curves:  
  
# km\_fit\_PFS <- survfit(Surv(time, status) ~ 1, data=df\_TTP)  
# plot(km\_fit\_PFS)  
#   
# km\_fit\_OS <- survfit(Surv(time, status) ~ 1, data=df\_TTD)  
# plot(km\_fit\_OS)  
  
# My understanding of survival curves is supported by: https://www.emilyzabor.com/tutorials/survival\_analysis\_in\_r\_tutorial.html  
  
#install.packages(c("survival", "lubridate", "ggsurvfit", "gtsummary", "tidycmprsk"))  
#remotes::install\_github("zabore/condsurv")  
#remotes::install\_github("zabore/ezfun")  
# library(survival)  
# library(lubridate)  
# library(ggsurvfit)  
# library(gtsummary)  
# library(tidycmprsk)  
# #library(condsurv)  
# #install.packages("magrittr") # package installations are only needed the first time you use it  
# #install.packages("dplyr") # alternative installation of the %>%  
# library(magrittr) # needs to be run every time you start R and want to use %>%  
# library(dplyr) # alternatively, this also loads %>%  
  
  
# We want to make sure the data we're feeding in matches the data from the publication in Table 4d and Supplementary Table 10d  
  
  
# A really good intro to survival curves is here: https://shariq-mohammed.github.io/files/cbsa2019/1-intro-to-survival.html  
  
  
survfit2(Surv(time, status) ~ 1, data = df\_TTP) %>% ggsurvfit() +  
 labs(  
 x = "Days",  
 y = "Progression Free survival probability"  
 ) +   
 add\_risktable()



km\_fit\_PFS <- survfit(Surv(time, status) ~ 1, data=df\_TTP)  
summary(km\_fit\_PFS, times = c(0,150,300,450, 600, 750, 900, 1050, 1200, 1350)) # This will stop just before the last patient is censored in the data, also, n.event means number of events that happened at this time.

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 0 202 0 1.0000 0.00000 1.00000 1.0000  
## 150 136 67 0.6683 0.03313 0.60644 0.7365  
## 300 65 69 0.3242 0.03303 0.26551 0.3958  
## 450 27 38 0.1347 0.02410 0.09482 0.1912  
## 600 11 15 0.0570 0.01656 0.03223 0.1007  
## 750 6 5 0.0311 0.01244 0.01418 0.0681  
## 900 3 3 0.0155 0.00888 0.00507 0.0477  
## 1050 1 1 0.0104 0.00728 0.00261 0.0410  
## 1200 1 0 0.0104 0.00728 0.00261 0.0410

survfit2(Surv(time, status) ~ 1, data = df\_TTD) %>% ggsurvfit() +  
 labs(  
 x = "Days",  
 y = "Overall survival probability"  
 ) +   
 add\_risktable()



km\_fit\_OS <- survfit(Surv(time, status) ~ 1, data=df\_TTD)  
summary(km\_fit\_OS, times = c(0,300,600,900,1200,1500,1800,1810,1820,1830,1840,1850,1860,1870,1880,1890,1891,1892,1893,1894,1895,1896,1897,1898,1899,1900,1901,2000,2010))

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 0 202 0 1.0000 0.0000 1.00000 1.000  
## 300 154 47 0.7667 0.0298 0.71047 0.827  
## 600 95 57 0.4809 0.0354 0.41638 0.555  
## 900 49 43 0.2581 0.0313 0.20340 0.327  
## 1200 15 24 0.1068 0.0242 0.06847 0.166  
## 1500 5 4 0.0725 0.0218 0.04023 0.131  
## 1800 1 1 0.0362 0.0278 0.00804 0.163  
## 1810 1 0 0.0362 0.0278 0.00804 0.163  
## 1820 1 0 0.0362 0.0278 0.00804 0.163  
## 1830 1 0 0.0362 0.0278 0.00804 0.163  
## 1840 1 0 0.0362 0.0278 0.00804 0.163  
## 1850 1 0 0.0362 0.0278 0.00804 0.163  
## 1860 1 0 0.0362 0.0278 0.00804 0.163  
## 1870 1 0 0.0362 0.0278 0.00804 0.163  
## 1880 1 0 0.0362 0.0278 0.00804 0.163  
## 1890 1 0 0.0362 0.0278 0.00804 0.163  
## 1891 1 0 0.0362 0.0278 0.00804 0.163  
## 1892 1 0 0.0362 0.0278 0.00804 0.163  
## 1893 1 0 0.0362 0.0278 0.00804 0.163  
## 1894 1 0 0.0362 0.0278 0.00804 0.163  
## 1895 1 0 0.0362 0.0278 0.00804 0.163  
## 1896 1 0 0.0362 0.0278 0.00804 0.163  
## 1897 1 0 0.0362 0.0278 0.00804 0.163  
## 1898 1 0 0.0362 0.0278 0.00804 0.163  
## 1899 1 0 0.0362 0.0278 0.00804 0.163

# The PFS and OS curves I create, match those from the publication, so we know the data we are feeding in is correct.  
  
# So, the problem must be in how I am generating probabilities in my model.  
  
  
  
# I calculate the probability of not progressing at the different time periods in the data from the publication, to contrast this to the probabilities I create of not progressing at different time periods once I've created probabilities in the R code, the probability of survival at each time period will be reported under the "survival" heading, per: https://www.emilyzabor.com/tutorials/survival\_analysis\_in\_r\_tutorial.html   
  
summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 0)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 0 202 0 1 0 1 1

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 150)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 150 136 67 0.668 0.0331 0.606 0.737

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 300)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 300 65 136 0.324 0.033 0.266 0.396

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 450)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 450 27 174 0.135 0.0241 0.0948 0.191

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 600)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 600 11 189 0.057 0.0166 0.0322 0.101

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 750)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 750 6 194 0.0311 0.0124 0.0142 0.0681

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 900)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 900 3 197 0.0155 0.00888 0.00507 0.0477

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 1050)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 1050 1 198 0.0104 0.00728 0.00261 0.041

summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 1200)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTP)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 1200 1 198 0.0104 0.00728 0.00261 0.041

# I can repeat this for TTD:  
  
  
summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 0)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 0 202 0 1 0 1 1

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 150)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 150 178 24 0.881 0.0228 0.838 0.927

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 300)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 300 154 47 0.767 0.0298 0.71 0.827

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 450)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 450 127 75 0.627 0.0341 0.564 0.698

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 600)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 600 95 104 0.481 0.0354 0.416 0.555

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 750)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 750 64 133 0.332 0.0335 0.272 0.404

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 900)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 900 49 147 0.258 0.0313 0.203 0.327

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 1050)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 1050 34 156 0.208 0.0294 0.157 0.274

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 1200)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 1200 15 171 0.107 0.0242 0.0685 0.166

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 1350)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 1350 10 173 0.0906 0.0231 0.0549 0.149

summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 1500)

## Call: survfit(formula = Surv(time, status) ~ 1, data = df\_TTD)  
##   
## time n.risk n.event survival std.err lower 95% CI upper 95% CI  
## 1500 5 175 0.0725 0.0218 0.0402 0.131

# Do the Weibull probabilities match the probabilities created directly by the Kaplan-Meier:  
  
# Here's how I checked if the Weibull probabilities match the probabilities created directly by the Kaplan-Meier, basically, I take the Kaplan-Meier probabilities at a few different cycles directly from the data as below, the probability of survival at each time period will be reported under the "survival" heading:  
  
# I do this first for TTP:  
  
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 0)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 14)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 28)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 42)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 56)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 70)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 84)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 98)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTP), times = 112)  
  
  
  
# Then, I calculate the probability of staying in the PFS state, which is what the above gives me. To do this, I create all the other probabilities as zero because I just want to look at whats going on with my Weibull probabilities for the PFS curve, I also put it in the first slot, as the way it works is that it needs to be multiplied by 1, and m\_M\_SoC has a cohort trace with 1 in in the first slot, ready to be matrix multiplied by m\_P\_SoC with it's probabilities. It's 1- because the probabilities created above from the PFS curve are probabilities of staying in PFS, not the probabilities of moving from PFS to OS.  
  
# m\_P\_SoC["PFS", "PFS",]<- (1 -p\_PFSOS\_SoC)  
# m\_P\_SoC["PFS", "OS",]<- 0  
# m\_P\_SoC["PFS", "Dead",]<-0  
#   
# # Setting the transition probabilities from OS  
# m\_P\_SoC["OS", "OS", ] <- 0  
# m\_P\_SoC["OS", "Dead", ] <- 0  
#   
#   
# # Setting the transition probabilities from Dead  
# m\_P\_SoC["Dead", "Dead", ] <- 0  
  
  
# So here I once again create the Markov cohort trace by looping over all cycles  
# - note that the trace can easily be obtained using matrix multiplications  
# - note that now the right probabilities for the cycle need to be selected, like I explained above. (this is all just the comments written from where I actually do the analysis).  
# for(i\_cycle in 1:(n\_cycle-1)) {  
# m\_M\_SoC[i\_cycle + 1, ] <- m\_M\_SoC[i\_cycle, ] %\*% m\_P\_SoC[ , , i\_cycle]  
# m\_M\_Exp[i\_cycle + 1, ] <- m\_M\_Exp[i\_cycle, ] %\*% m\_P\_Exp[ , , i\_cycle]  
# }  
  
  
# head(m\_M\_SoC) # print the first few lines of the matrix for standard of care (m\_M\_SoC)  
# head(m\_M\_Exp) # print the first few lines of the matrix for experimental treatment(m\_M\_Exp)  
  
# looking at the cohort trace -> m\_M\_SoC, you see that the proportions in the PFS state from 100 are basically identical to the probabilities of being in those states created from the Kaplan Meier, again starting from 100% of people under study, thus, the Weibull probabilities created match the Kaplan-Meier.  
  
# They can be a few % off but that's OK, as it's fitting a Weibull to the Kaplan Meier curves, so they're not going to be identical.  
  
# m\_M\_SoC  
  
  
# I can repeat this for TTD:  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 0)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 14)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 28)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 42)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 56)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 70)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 84)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 98)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 112)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 126)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 158)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 172)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 184)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 198)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 212)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 224)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 238)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 252)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 266)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 280)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 294)  
#   
# summary(survfit(Surv(time, status) ~ 1, data = df\_TTD), times = 308)  
  
  
# Here, like above, I put the p\_PFSD\_SoC bit into the first part of the probability matrix, even though it concerns people going from first line treatment or PFS to dead, because it needs to be multiplied by the 1 that exists in the cohort:  
  
#   
# m\_P\_SoC["PFS", "PFS",]<- 1- p\_PFSD\_SoC  
# m\_P\_SoC["PFS", "OS",]<- 0  
# m\_P\_SoC["PFS", "Dead",]<-0  
#   
# # Setting the transition probabilities from OS  
# m\_P\_SoC["OS", "OS", ] <- 0  
# m\_P\_SoC["OS", "Dead", ] <- 0  
#   
#   
# # Setting the transition probabilities from Dead  
# m\_P\_SoC["Dead", "Dead", ] <- 0  
  
  
# So here I once again create the Markov cohort trace by looping over all cycles  
# - note that the trace can easily be obtained using matrix multiplications  
# - note that now the right probabilities for the cycle need to be selected, like I explained above.  
# for(i\_cycle in 1:(n\_cycle-1)) {  
# m\_M\_SoC[i\_cycle + 1, ] <- m\_M\_SoC[i\_cycle, ] %\*% m\_P\_SoC[ , , i\_cycle]  
# m\_M\_Exp[i\_cycle + 1, ] <- m\_M\_Exp[i\_cycle, ] %\*% m\_P\_Exp[ , , i\_cycle]  
# }  
#   
#   
# head(m\_M\_SoC) # print the first few lines of the matrix for standard of care (m\_M\_SoC)  
# head(m\_M\_Exp) # print the first few lines of the matrix for experimental treatment(m\_M\_Exp)  
#   
# #m\_M\_SoC  
# m\_M\_SoC  
  
# This also answers the question of how exactly probabilities are created, the m\_P\_SoC is something that, when multiplied by 1, (when at the start 100% of our cohort are in the PFS state) gives us the first probability from that 100%, and the next part or row of that m\_P\_SoC probability is then multiplied by the proportion who are in the different states in the next row for the cohort (given their movements last time having been multiplied by m\_P\_SoC) which gives us the second probability multiplied by the proportion of that 1 (or 100%) in each state, and so on, and so forth, for each wave.

# Time-to-Progression (TTP):

# 04 Parametric Survival Analysis itself:

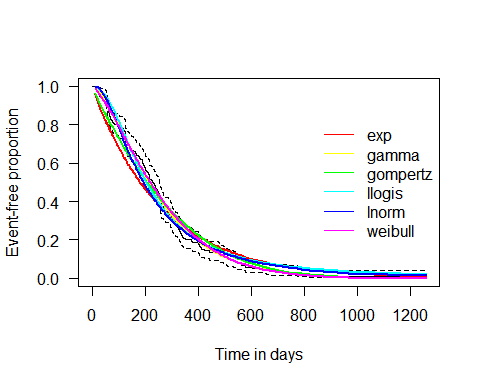
# We use the 'flexsurv' package to fit several commonly used parametric survival distributions.  
  
# The data needs to be set up to include a column for the time (in years) and a status indicator whether the time corresponds to an event, i.e. progression (status = 1), or to the last time of follow up, i.e. censoring (status = 0).  
  
  
# It looks like Koen is applying the flexsurvreg formula to individuals who experience progression (i.e. ~1):  
  
head(df\_TTP)

## time status  
## 1 739 1  
## 2 211 1  
## 3 311 1  
## 4 412 1  
## 5 25 1  
## 6 302 1

l\_TTP\_SoC\_exp <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTP, dist = "exp")  
l\_TTP\_SoC\_gamma <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTP, dist = "gamma")  
l\_TTP\_SoC\_gompertz <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTP, dist = "gompertz")  
l\_TTP\_SoC\_llogis <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTP, dist = "llogis")  
l\_TTP\_SoC\_lnorm <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTP, dist = "lnorm")  
l\_TTP\_SoC\_weibull <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTP, dist = "weibull")

# 05 Inspecting the fits:

# And this would make sense per the below diagram - which looks at the proportion of individuals who have the event, i.e. progression.  
  
# Inspect fit based on visual fit  
colors <- rainbow(6)  
plot(l\_TTP\_SoC\_exp, col = colors[1], ci = FALSE, ylab = "Event-free proportion", xlab = "Time in days", las = 1)  
lines(l\_TTP\_SoC\_gamma, col = colors[2], ci = FALSE)  
lines(l\_TTP\_SoC\_gompertz, col = colors[3], ci = FALSE)  
lines(l\_TTP\_SoC\_llogis, col = colors[4], ci = FALSE)  
lines(l\_TTP\_SoC\_lnorm, col = colors[5], ci = FALSE)  
lines(l\_TTP\_SoC\_weibull, col = colors[6], ci = FALSE)  
legend("right",  
 legend = c("exp", "gamma", "gompertz", "llogis", "lnorm", "weibull"),  
 col = colors,  
 lty = 1,  
 bty = "n")



# Koen says "# Weibull has the best visual and numerical fit" but I don't see what it's visually being compared to in this graph, I will have to learn about this in the C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\flexsurv folder.  
  
# The time is in years which makes sense, the data you are drawing from is in years so that's what you want to make comparisons to.

# Compare the fit numerically based on the AIC  
(v\_AIC <- c(  
 exp = l\_TTP\_SoC\_exp$AIC,  
 gamma = l\_TTP\_SoC\_gamma$AIC,  
 gompertz = l\_TTP\_SoC\_gompertz$AIC,  
 llogis = l\_TTP\_SoC\_llogis$AIC,  
 lnorm = l\_TTP\_SoC\_lnorm$AIC,  
 weibull = l\_TTP\_SoC\_weibull$AIC  
))

## exp gamma gompertz llogis lnorm weibull   
## 2604.383 2570.365 2594.880 2579.625 2580.161 2574.970

# Weibull has the best visual and numerical fit  
  
# I will have to learn what a good value of AIC is, and, because flexsurvreg provides AIC I will have to see what other numerical measures, such as BIC, it provides.

# 06 Saving the survival parameters for use in the model:

# Saving the survival parameters ----  
  
# The 'flexsurv' package return the coefficients, which need to be transformed for use in the base R functions, but that will be done when the coefficients actually are used, for the time being we will just save the survival parameters from the distribution we decide to use.   
  
# NB, if we are not going with Weibull then we may have to save something specific to the distribution that is not shape or scale - we can look into this if we don't use Weibull.  
  
l\_TTP\_SoC\_weibull

## Call:  
## flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTP,   
## dist = "weibull")  
##   
## Estimates:   
## est L95% U95% se   
## shape 1.3917 1.2520 1.5470 0.0751  
## scale 286.7776 258.1993 318.5191 15.3598  
##   
## N = 202, Events: 198, Censored: 4  
## Total time at risk: 52049  
## Log-likelihood = -1285.485, df = 2  
## AIC = 2574.97

# Calling a flexsurvreg parameter like this allows you to see here that Weibull is the shape and the scale, so if we do go with another distribution we can see what it's version of shape and scale are and use these instead.  
  
l\_TTP\_SoC\_weibull$coefficients

## shape scale   
## 0.3305319 5.6587072

coef\_weibull\_shape\_SoC <- l\_TTP\_SoC\_weibull$coefficients["shape"]  
coef\_weibull\_scale\_SoC <- l\_TTP\_SoC\_weibull$coefficients["scale"]

# Time-to-Dead (TTD):

# 07 Parametric Survival Analysis itself:

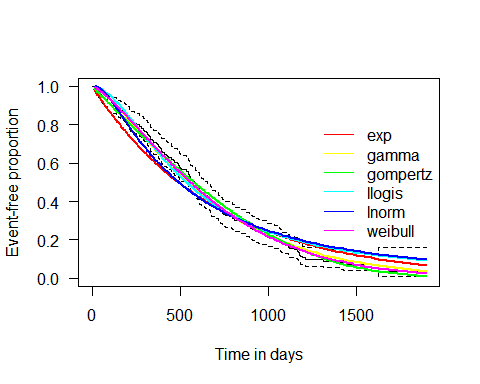
# We use the 'flexsurv' package to fit several commonly used parametric survival distributions.  
  
# The data needs to be set up to include a column for the time (in years) and a status indicator whether the time corresponds to an event, i.e. progression (status = 1), or to the last time of follow up, i.e. censoring (status = 0).  
  
  
# It looks like Koen is applying the flexsurvreg formula to individuals who experience progression (i.e. ~1):  
  
# I duplicate the time to progression data frame as time to dead here, I'll REALLY need to make sure to delete this line when I have actual time to dead data or else I'll be running my analysis on the time to progression data twice and thinking I'm running it on time to dead:  
  
# df\_TTD <- df\_TTP  
  
head(df\_TTD)

## time status  
## 1 1899 0  
## 2 211 1  
## 3 610 1  
## 4 673 1  
## 5 25 1  
## 6 1187 1

l\_TTD\_SoC\_exp <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTD, dist = "exp")  
l\_TTD\_SoC\_gamma <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTD, dist = "gamma")  
l\_TTD\_SoC\_gompertz <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTD, dist = "gompertz")  
l\_TTD\_SoC\_llogis <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTD, dist = "llogis")  
l\_TTD\_SoC\_lnorm <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTD, dist = "lnorm")  
l\_TTD\_SoC\_weibull <- flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTD, dist = "weibull")

# 08 Inspecting the fits:

# And this would make sense per the below diagram - which looks at the proportion of individuals who have the event, i.e. progression.  
  
# Inspect fit based on visual fit  
colors <- rainbow(6)  
plot(l\_TTD\_SoC\_exp, col = colors[1], ci = FALSE, ylab = "Event-free proportion", xlab = "Time in days", las = 1)  
lines(l\_TTD\_SoC\_gamma, col = colors[2], ci = FALSE)  
lines(l\_TTD\_SoC\_gompertz, col = colors[3], ci = FALSE)  
lines(l\_TTD\_SoC\_llogis, col = colors[4], ci = FALSE)  
lines(l\_TTD\_SoC\_lnorm, col = colors[5], ci = FALSE)  
lines(l\_TTD\_SoC\_weibull, col = colors[6], ci = FALSE)  
legend("right",  
 legend = c("exp", "gamma", "gompertz", "llogis", "lnorm", "weibull"),  
 col = colors,  
 lty = 1,  
 bty = "n")



# Koen says "# Weibull has the best visual and numerical fit" but I don't see what it's visually being compared to in this graph, I will have to learn about this in the C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\flexsurv folder.  
  
# The time is in years which makes sense, the data you are drawing from is in years so that's what you want to make comparisons to.

# Compare the fit numerically based on the AIC  
(v\_AIC <- c(  
 exp = l\_TTD\_SoC\_exp$AIC,  
 gamma = l\_TTD\_SoC\_gamma$AIC,  
 gompertz = l\_TTD\_SoC\_gompertz$AIC,  
 llogis = l\_TTD\_SoC\_llogis$AIC,  
 lnorm = l\_TTD\_SoC\_lnorm$AIC,  
 weibull = l\_TTD\_SoC\_weibull$AIC  
))

## exp gamma gompertz llogis lnorm weibull   
## 2664.472 2648.915 2648.350 2665.054 2680.234 2646.087

# Weibull has the best visual and numerical fit  
  
# I will have to learn what a good value of AIC is, and, because flexsurvreg provides AIC I will have to see what other numerical measures, such as BIC, it provides.

# 09 Saving the survival parameters for use in the model:

# Saving the survival parameters ----  
  
# The 'flexsurv' package return the coefficients, which need to be transformed for use in the base R functions, but that will be done when the coefficients actually are used, for the time being we will just save the survival parameters from the distribution we decide to use.   
  
# NB, if we are not going with Weibull then we may have to save something specific to the distribution that is not shape or scale - we can look into this if we don't use Weibull.  
  
l\_TTD\_SoC\_weibull

## Call:  
## flexsurvreg(formula = Surv(time, status) ~ 1, data = df\_TTD,   
## dist = "weibull")  
##   
## Estimates:   
## est L95% U95% se   
## shape 1.3530 1.1967 1.5297 0.0847  
## scale 733.3499 656.5785 819.0978 41.3754  
##   
## N = 202, Events: 176, Censored: 26  
## Total time at risk: 124782  
## Log-likelihood = -1321.043, df = 2  
## AIC = 2646.087

# Calling a flexsurvreg parameter like this allows you to see here that Weibull is the shape and the scale, so if we do go with another distribution we can see what it's version of shape and scale are and use these instead.  
  
l\_TTD\_SoC\_weibull$coefficients

## shape scale   
## 0.302343 6.597623

coef\_TTD\_weibull\_shape\_SoC <- l\_TTD\_SoC\_weibull$coefficients["shape"]  
coef\_TTD\_weibull\_scale\_SoC <- l\_TTD\_SoC\_weibull$coefficients["scale"]

**How to extrapolate the parametric models beyond the KM curve**

1 How to deal with time in parametric survival analysis, that is, when I digitise for a certain period of time, how do I convert it to the time period I want to look at in my own analysis? [there may be a clue to this by how they handle time here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\flexsurv\THE FLEXSURV EXAMPLE CODE EXPLAINED.txt OS\_TimeYears<-AVAL\*(1/12) it looks like they wanted overal survival time in years, and so converted AVAL by 1/12]

2 How to extrapolate the parameteric models beyond the KM curve they came from.

3 What does the flexsurvreg code mean?

Use the below (and above quote section) for digitising, engauge, parametric survival modelling, etc., There is helpful R code on doing this here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\ISPOR WEBINAR Health Economic Modelling in R and I think I should go through all the R code I have with a fine toothcomb to look for further resources in R on how to do this. Because this R code specificaly has: “# 2. SURVIVAL ANALYSIS survival analysis of individual patient data”

Read the below in tangent with:

Cost-Effectiveness Analysis of Regorafenib for Metastatic Colorectal Cancer Daniel A. Goldstein, Bilal B. Ahmad, Qiushi Chen, Turgay Ayer, David H. Howard, Joseph Lipscomb, Bassel F. El-Rayes, and Christopher R. Flowers <https://sci-hub.se/10.1200/JCO.2015.61.9569>

and also this article: Peng, Z., Hou, X., Huang, Y., Xie, T., & Hua, X. (2020). Cost-effectiveness analysis of fruquintinib for metastatic colorectal cancer third-line treatment in China. BMC cancer, 20(1), 1-8. <https://link.springer.com/article/10.1186/s12885-020-07486-w>

and:

Ewara, E. M., Zaric, G. S., Welch, S., & Sarma, S. (2014). Cost-effectiveness of first-line treatments for patients with KRAS wild-type metastatic colorectal cancer. Current Oncology, 21(4), 541-550. <https://sci-hub.st/10.3747/co.21.1837>

Progression Risk and Mortality Progression risks and cause-specific mortalities for each arm were derived from the PFS and OS curves of the corresponding study. Engauge Digitizer16 was used to extract the data points from each PFS and OS plot, and then used to fit parametric survival models. Statistical analyses demonstrated that Weibull and log-logistic models provided a good fit for all curves according to the Akaike information criterion, Schwarz Bayesian criterion,17 and visual inspection (Supplemental Table 1 in the online version). Weibull models permit the hazard rate to increase over time and are thus more suitable for modeling events occurring early during follow-up periods. We therefore selected the Weibull model for all survival curves in this analysis. Next, we estimated the risk for each cycle based on the fitted survival models. In particular, because OS is defined as the period between the start of treatment and death, we computed the cause-specific mortality rates, p1(t), at cycle t based on the fitted Weibull OS model, OS(t), using the formula:

p1ðtÞ ¼ Pðt T t þ 1Þ PðT tÞ ¼ ðOSðtÞ OSðt 1ÞÞ OSðtÞ : (1)

PFS is the period between the start and progression of disease or death. We computed the combined risks p2(t) based on the Weibull PFS model using Formula 1 and computed p2(t) p1(t) as the estimate of the progression risks. Estimates of mortality and progression risk beyond the follow-up time in the clinical trials were extrapolated based on the fitted survival models. The overall mortality was defined according to the maximum value of cause-specific mortality and the background mortality. We use the US life tables to estimate the background mortality for each age group separately.18 **In each run of the model simulation, the initial age was sampled from the distribution of age at diagnosis for mCRC in the Surveillance, Epidemiology, and End Results database since the year 2000. [So, rather than picking a starting age, it looks like they randomly sampled an age for competing background mortality].**

Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., … & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. *Clinical colorectal cancer*, *13*(4), 219-225.

I discussed parametric survival analysis with Andy here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A1\_Advanced Markov Modelling\A1.3 Practical exercise R

**[So, I think my takeway is that I need to do digitising and maybe parametric survival analysis on some existing published data - per C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Parametric Survival Analysis and the rest of my York training material THEN I’ll need to include Age specific all cause mortality per: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH because I don’t have my own clinical data, these are both good sources to consider seeing as they also didnt have their own data, I’ll also have to think about the age of the cohort if I include competing mortality, as below:].**

**All done below:**

*“Starting age of the cohort was 60 years, which is the median age of incident breast cancer cases as reported by The Netherlands Cancer Registration [28].” per: Kip, M., Monteban, H., & Steuten, L. (2015). Long-term cost–effectiveness of Oncotype DX® versus current clinical practice from a Dutch cost perspective. Journal of comparative effectiveness research, 4(5), 433-445. C:/Users/Jonathan/Dropbox/PC/Downloads/kip2015%20(1).pdf*

*“Nonbreast cancer death was captured as a competing risk in the model. These data were derived from The Netherlands Cancer Registration and based on Dutch female life tables for 2007–2009 [28,33]. Kip, M., Monteban, H., & Steuten, L. (2015). Long-term cost–effectiveness of Oncotype DX® versus current clinical practice from a Dutch cost perspective. Journal of comparative effectiveness research, 4(5), 433-445.”* [*file:///C:/Users/Jonathan/Dropbox/PC/Downloads/kip2015%20(1).pdf*](file:///C:/Users/Jonathan/Dropbox/PC/Downloads/kip2015%20(1).pdf)

*Patients had a starting age of 56 years Salem, A., Men, P., Ramos, M., Zhang, Y. J., Ustyugova, A., & Lamotte, M. (2021). Cost–effectiveness analysis of empagliflozin compared with glimepiride in patients with Type 2 diabetes in China. Journal of Comparative Effectiveness Research, 10(6), 469-480.*

#### Utility Estimates

To present health outcomes in terms of QALYS, we adjusted survival time by quality of life using a Health Related Quality of Life (HR-QOL) score on a 0-1 utility scale. The quality of life utilities of a cycle spent in each of the model health states has been estimated to be 0.85 for the first line health state, and 0.65 for the second line health state in the published literature on quality of life utility of patients with mCRC without complications (Ramsey et al. 2000) FIND NEW UTILITIES ELSEWHERE [We want to be clear, per Craig, D., McDaid, C., Fonseca, T., Stock, C., Duffy, S., & Woolacott, N. (2009). Are adverse effects incorporated in economic models? An initial review of current practice. *Health Technology Assessment (Winchester, England)*, *13*(62), 1-71., that our utilities come from patients without adverse events, otherwise the disutilty of these adverse events have already been counted, so we want utility before any adverse events occur, and then we can decrease this with disutility for these adverse events per the studies AND MAYBE EXPLAIN MY PLAN WITH KATHLEEN HERE TO GET COUNTRY SPECIFIC UTILITY VALUES]. We assumed in the model that utility was only related to health state, and not to therapy.

#### Probability of Adverse events:

We established the probability of Grade 3 and 4 adverse events (AEs) from the trials used for the models.15,20 Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., … & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. *Clinical colorectal cancer*, *13*(4), 219-225.

To do this, they basically went to the studies by:

Capitain et al, 2012

Hochster et al, 2008

Bennouna et al, 2013

And went to the tables on “Incidence of Grade X and Grade Y Adverse Events”, and took the percentages reported in these studies to use in their own table under “Adverse Event Incidence”

Then, according to: [[https://en.wikipedia.org/wiki/Incidence\_(epidemiology](https://en.wikipedia.org/wiki/Incidence\_(epidemiology))](https://en.wikipedia.org/wiki/Incidence_(epidemiology)) because we have explicitly an “incidence” we are allowed use this as a probability in our analysis, so, having filled in the “Adverse Event Incidence” section in our Table on model parameters like they do, we can then use these “incidence” values as probabilities.

Remember that Leukopenia = Febrile Neutropenia.

Adverse events were added as a health state in the Markov model, per (Nebuloni et al. 2013) (Goulart and Ramsey 2011a). The probability, and disutility, of Grade 3/4 adverse events (AEs) was informed by the literature (Hochster et al. 2008) (Aballéa et al. 2007). [PROBABBILITY:Leukopenia = 0.040 Diarrhea = 0.310 Vomiting = 0.310, –> POTENTIALLY I SHOULD TAKE THE PROBABILITIES UNDER BVZ FOR AE’S FROM HERE TOO AND USE THEM IN THE MODEL. DISUTILITY:Leukopenia = 45% LOWER PFS UTILITY Diarrhea = 36% LOWER PFS UTILITY Vomiting = 19% LOWER PFS UTILITY]. Duration-adjusted disutility was subtracted from the baseline utility to calculate the overall utility of each health state, we assumed the duration of each adverse event was 7 days, and that treatment-induced complications are mutually exclusive (Goldstein, Ahmad, et al. 2015) (Refaat et al. 2014).

*AEs can reduce patients’ utility so we applied disutility estimates for these temporary health states to each major AE based on data from Aballea at al.21 These included febrile neutropenia, vomiting, and diarrhea. We assumed the duration of each AE was 5 days. For each AE, the average disutility was weighted by the incidence reported in the clinical study.*

*Disutilities associated with adverse events (AEs), including fatigue, hand-foot syndrome, diarrhea, and hypertension, were obtained from published data in the literature13. The duration of AEs was estimated based on Expert Consultation. From the opinions of many doctors in China, regardless of patients in the arm of fruquintinib or regorafenib, hand-foot syndrome would last for 14 days and the disutility was 0.116. Hypertension would last for five days and the disutility was 0. Diarrhea would last for five days and the disutility was 0.1038 . The duration-adjusted disutility was subtracted from the baseline utility to calculate the overall utility of each health state.*

Baseline quality of life utility of Chinese patients with T2D without complications was 0.876 [[**19**](https://www.futuremedicine.com/reader/content/17e4aa15614/10.2217/cer-2020-0284/format/epub/EPUB/xhtml/index.xhtml#B19)] and BMI-related disutility applied per BMI point increase was 0.0061 [[**20**](https://www.futuremedicine.com/reader/content/17e4aa15614/10.2217/cer-2020-0284/format/epub/EPUB/xhtml/index.xhtml#B20)]. For acute events, a disutility was subtracted from the base utility, leading to a final utility score recorded for the patient experiencing the event. In case, a patient suffered different multiple events (e.g., a myocardial infarction [MI] and a stroke), the model followed the minimum approach, recording the utility value of the health state with the lowest individual score, while still applying a disutility related to adverse events. The full list of utilities and disutilities can be retrieved from [**Supplementary Table 2**](https://www.futuremedicine.com/doi/suppl/10.2217/cer-2020-0284).

Salem, A., Men, P., Ramos, M., Zhang, Y. J., Ustyugova, A., & Lamotte, M. (2021). Cost–effectiveness analysis of empagliflozin compared with glimepiride in patients with Type 2 diabetes in China. Journal of Comparative Effectiveness Research, 10(6), 469-480.

*If a patient suffered multiple adverse events, I think they just applied the disutility that was the worst to their baseline utility, rather than trying to combine disutilities - which would probably be messy.*

*We assumed that treatment-induced complications—grade 3 or 4—are mutually exclusive—a patient can have only 1 complication at a time*

*Refaat, T., Choi, M., Gaber, G., Kiel, K., Mehta, M., Gradishar, W., & Small Jr, W. (2014). Markov model and cost-effectiveness analysis of bevacizumab in HER2-negative metastatic breast cancer. American journal of clinical oncology, 37(5), 480-485.* [*https://sci-hub.se/10.1097/COC.0b013e31827e4e9a*](https://sci-hub.se/10.1097/COC.0b013e31827e4e9a)

*This is an alternative approach that may be cleaner, assume patients can only have one adverse event per cycle, and that way you don’t have to find out which is worse and apply that one.*

*Lots of adverse events in studies to consider here:*

*“The frequencies of occurrence of AEFIs were expressed through probabilities, incidence, rates, or relative risk mentioned in 35 of the included studies [27–34,36,38– 46,49,51,52,55–57,59,61,63,64,68,72,73,75–78].”*

*- Fens, T., de Boer, P. T., van Puijenbroek, E. P., & Postma, M. J. (2021). Inclusion of Safety-Related Issues in Economic Evaluations for Seasonal Influenza Vaccines: A Systematic Review. Vaccines 2021, 9, 111.*

[*https://sci-hub.se/10.3390/vaccines9020111*](https://sci-hub.se/10.3390/vaccines9020111)

###### *Adverse Events Occurring:*

If I’m made to include adverse events for second line therapy, which I don’t think is necessary as second line therapy is the same for everyone, information on the incidence of adverse events is here for Folfiri on it’s own, or in combination with Bevacizumab: Wong, W. W., Zargar, M., Berry, S. R., Ko, Y. J., Riesco-Martínez, M., & Chan, K. K. W. (2019). Cost-effectiveness analysis of selective first-line use of biologics for unresectable RAS wild-type left-sided metastatic colorectal cancer. Current Oncology, 26(5), 597-609. <file:///C:/Users/Jonathan/Dropbox/Grammarly%20files/curroncol-26-04843.pdf>

*Have a look at my emails to Kathleen and Daniel re: Adverse events, utilities during adverse events, etc.,*

*“The frequencies of occurrence of AEFIs were expressed through probabilities, incidence, rates, or relative risk mentioned in 35 of the included studies [27–34,36,38– 46,49,51,52,55–57,59,61,63,64,68,72,73,75–78].”*

*- Per:*

*Fens, T., de Boer, P. T., van Puijenbroek, E. P., & Postma, M. J. (2021). Inclusion of Safety-Related Issues in Economic Evaluations for Seasonal Influenza Vaccines: A Systematic Review. Vaccines 2021, 9, 111.*

[*https://sci-hub.se/10.3390/vaccines9020111*](https://sci-hub.se/10.3390/vaccines9020111)

To convert reported incidence rates to probabilities read:

<https://sphweb.bumc.bu.edu/otlt/mph-modules/ep/ep713_diseasefrequency/EP713_DiseaseFrequency5.html>

Also saved here:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A1\_Advanced Markov Modelling\A1.1.3 Key concepts in time-dependent transition probabilities\Relationship of Incidence Rate to Cumulative Incidence (Risk).pdf

The HERC Material here:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\HERC Cost Effectiveness Analysis Course\Estimating Transition Probabilities

Probability converting is here per the York course:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A1\_Advanced Markov Modelling\A1.1.3 Key concepts in time-dependent transition probabilities\notes

This may also be helpful

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Foundation Course\F2\_Decision Trees\F2.1.2 Probabilities & conditioning\Probabilities and Conditioning Notes

*These guys add adverse events as a State in the Markov model, so I could probably just do that:*

*Nebuloni, D. R., Mak, M. P., Souza, F. H., Saragiotto, D. F., Júlio, T., De Castro Jr, G., … & Hoff, P. M. (2013). Modified FLOX as first-line chemotherapy for metastatic colorectal cancer patients in the public health system in Brazil: Effectiveness and cost-utility analysis. Molecular and Clinical Oncology, 1(1), 175-179.* [*https://sci-hub.st/10.3892/mco.2012.12*](https://sci-hub.st/10.3892/mco.2012.12)

*Goulart, B., & Ramsey, S. (2011). A trial-based assessment of the cost-utility of bevacizumab and chemotherapy versus chemotherapy alone for advanced non-small cell lung cancer. Value in Health, 14(6), 836-845.* [*<https://sci-hub.se/10.1016/j.jval.2011.04.004>*](https://sci-hub.se/10.1016/j.jval.2011.04.004)

*The duration-adjusted disutility was subtracted from the baseline utility to calculate the overall utility of each health state.*

*Cost-Effectiveness Analysis of Regorafenib for Metastatic Colorectal Cancer Daniel A. Goldstein, Bilal B. Ahmad, Qiushi Chen, Turgay Ayer, David H. Howard, Joseph Lipscomb, Bassel F. El-Rayes, and Christopher R. Flowers* [*https://sci-hub.se/10.1200/JCO.2015.61.9569*](https://sci-hub.se/10.1200/JCO.2015.61.9569)

*Progression Risk and Mortality Progression risks and cause-specific mortalitie*

###### Range around things:

The below, and most of Table 4 in Goldstein, Daniel A., Qiushi Chen, Turgay Ayer, David H. Howard, Joseph Lipscomb, R. Donald Harvey, Bassel F. El-Rayes, and Christopher R. Flowers. 2014. “Cost Effectiveness Analysis of Pharmacokinetically-Guided 5-Fluorouracil in FOLFOX Chemotherapy for Metastatic Colorectal Cancer.” Clinical Colorectal Cancer 13 (4): 219–25. [<https://doi.org/10.1016/j.clcc.2014.09.007.>](https://doi.org/10.1016/j.clcc.2014.09.007.)

seems to suggest I can determine my upper and lower limit simply as 20% of my point estimate, most, if not all, of the time. Also, Goldstein take the 20% approach and apply this to the Beta distribution for utilities, so it must be possible to apply to both the Beta distribution and the normal distribution application I saw in the study from Eygpt.

“All model inputs were varied 20% below and above the base case for costs, and confidence intervals or standard errors extracted from published sources were used for the probabilities, adverse events, and utilities (Table 1).”

The transition probabilities had a max of 20% over the point estimate, a min of 20% under the point estimate, and a Gamma distribution for transition probabilities.

- Per: Hamdy Elsisi, G., Nada, Y., Rashad, N., & Carapinha, J. (2019). Cost-effectiveness of sorafenib versus best supportive care in advanced hepatocellular carcinoma in Egypt. Journal of Medical Economics, 22(2), 163-168. [<https://www-tandfonline-com.proxy.library.rcsi.ie/doi/pdf/10.1080/13696998.2018.1552432>](https://www-tandfonline-com.proxy.library.rcsi.ie/doi/pdf/10.1080/13696998.2018.1552432)

The below takes the max and min as 20% of the mean for costs, utility, applying a gamma and beta distribution respectively (negative beta for disutilities that were negative percentages). For the discount factor it looks like they just took a minimum of no discount factor and a maximum of 5% and applied a uniform distribution.

Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., … & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. *Clinical colorectal cancer*, *13*(4), 219-225.

***I like that Goldstein et al., report the actual distribution they derived from their Value, Minimum and Maximum, I think I’ll review the other papers by Goldstein and co-authors to see if they repeat that elsewhere, as it supports checking your methods against their own.***

Transition probabilities used in the analysis

The transition probabilities, had a min 20% below the base case value and a max 20% above this.

Wu, Q., Wang, X., Zhang, M., Liao, W., Wang, F., & Li, Q. (2020). Cost-effectiveness analysis of bevacizumab plus paclitaxel versus bevacizumab plus capecitabine for HER2-negative locally recurrent or metastatic breast cancer. Oncology Research and Treatment, 43(4), 153-159. [<https://www.karger.com/Article/Pdf/505932>](https://www.karger.com/Article/Pdf/505932)

20% costs variation here: Sharp, L., Tilson, L., Whyte, S., O’Ceilleachair, A., Walsh, C., Usher, C., … & Comber, H. (2012). Cost-effectiveness of population-based screening for colorectal cancer: a comparison of guaiac-based faecal occult blood testing, faecal immunochemical testing and flexible sigmoidoscopy. British journal of cancer, 106(5), 805-816. <https://www.nature.com/articles/bjc2011580.pdf>

20% costs variability Goulart, B., & Ramsey, S. (2011). A trial-based assessment of the cost-utility of bevacizumab and chemotherapy versus chemotherapy alone for advanced non-small cell lung cancer. *Value in Health*, *14*(6), 836-845.

**Costing Data in Spain:**

Rivera, F., Valladares, M., Gea, S., & López-Martínez, N. (2017). Cost-effectiveness analysis in the Spanish setting of the PEAK trial of panitumumab plus mFOLFOX6 compared with bevacizumab plus mFOLFOX6 for first-line treatment of patients with wild-type RAS metastatic colorectal cancer. Journal of Medical Economics, 20(6), 574-584. <https://sci-hub.st/10.1080/13696998.2017.1285780>

#### Sensitivity Analysis

I should do this in tangent with the two papers using 20%.

{# {r setup, include=FALSE} # knitr::opts\_chunk$set(echo = TRUE, warning = FALSE, message = FALSE, eval = T) # # To knit this document every time it is run, you can change `eval` to `TRUE` in the above.

# rm(list = ls())   
# clear memory (removes all the variables from the work space)

# 01 Load packages

{# {r} # if (!require('pacman')) install.packages('pacman'); library(pacman) # # use this package to conveniently install other packages # # load (install if required) packages from CRAN # p\_load("diagram", "dampack", "reshape2") # # library(devtools) # devtools is necessary to install from github. # # install\_github("DARTH-git/darthtools", force = TRUE) # Uncomment if there is a newer version # p\_load\_gh("DARTH-git/darthtools")

# 02 Load functions

# all functions are in the darthtools package  
  
# There is a functions RMD for the PSA stuff below, instead of calling it in: ## 08.2 Load PFS-PFSer Markov model function, I could just place it here, and place all the necessary packages above and then I would only ever need 1 R Markdown document for the entire study. Will think about doing this.

# To use dampack, review this before you start:

<https://cran.r-project.org/web/packages/dampack/vignettes/basic_cea.html>

<https://cran.r-project.org/web/packages/dampack/vignettes/dsa_generation.html>

<https://cran.r-project.org/web/packages/dampack/vignettes/psa_generation.html>

<https://cran.r-project.org/web/packages/dampack/vignettes/psa_analysis.html>

<https://cran.r-project.org/web/packages/dampack/vignettes/voi.html>

<https://cran.r-project.org/web/packages/dampack/dampack.pdf>

<https://syzoekao.github.io/CEAutil/#43_one-way_sensitivity_analysis>

<https://hesim-dev.github.io/hesim/articles/cea.html>

also saved as pdf files here:

C:- Royal College of Surgeons in IrelandCode\_Model

And:

“An Introductory Tutorial on Cohort State-Transition Models in R Using a Cost-Effectiveness Analysis” <https://arxiv.org/pdf/2001.07824.pdf> [also saved here: C:- Royal College of Surgeons in IrelandCode\_ModelIntroductory Tutorial on Cohort StateTransition.pdf]

The dampack package was based on the following textbook:

<file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/R%20Code/GitHub/COLOSSUS_Model/(Cambridge%20medicine)%20Hunink,%20M.%20G.%20Myriam_Weinstein,%20Milton%20C%20-%20Decision%20making%20in%20health%20and%20medicine_%20integrating%20evidence%20and%20values.pdf>

If I find any parts of the package confusing I can email:

Maintainer: Greg Knowlton <knowl193@umn.edu>

The official website is here: <https://cran.r-project.org/web/packages/dampack/>

IT’S ALSO VERY IMPORTANT TO CONSIDER WHAT BEARING THE INCLUSION OF ADDITIONAL MARKOV BUBBLES WILL HAVE ON THE FUNCTIONS USED HERE, WHICH ASSUMEDLY WERE BUILT WITH A 3 STATE MODEL IN MIND, AND NOW WE HAVE ADDITIONAL TUNNEL STATES.

# 03 Input model parameters

Great utility values for first and second line treatment and adverse events here: Cost-effectiveness analysis of selective first-line use of biologics for unresectable RAS wild-type left-sided metastatic colorectal cancer saved here: <file:///C:/Users/Jonathan/Dropbox/Grammarly%20files/curroncol-26-04843.pdf>

## General setup  
  
  
# Why ordering, V\_names\_states v\_tc\_SoC, v\_tc\_Exp, v\_tu\_SoC and v\_tu\_Exp matters:  
  
# The ordering of V\_names\_states has an influence on the tornado diagram and the reported cost-effectiveness results.  
  
# So, I need to ensure I correctly order V\_names\_states all the way through from the start.  
  
# m\_P\_Exp and m\_P\_SoC both use v\_names\_states to set up the names of the rows and columns of their matrices.   
  
# As does m\_M\_SoC and m\_M\_Exp.  
  
# Then m\_M\_SoC and m\_M\_Exp use the row column names to fill in the 100% of the cohort (or 1) in PFS and the 0% of the cohort in AE1, AE2, AE3, OS and DEAD in wave 1. m\_P\_Exp and m\_P\_SoC also both use the row and column names to fill in the transition probabilities between PFS and OS, etc.,  
  
# Then m\_M\_SoC and m\_M\_Exp are matrix multiplied by m\_P\_Exp and m\_P\_SoC:  
  
# for(i\_cycle in 1:(n\_cycle-1)) {  
# m\_M\_SoC[i\_cycle + 1, ] <- m\_M\_SoC[i\_cycle, ] %\*% m\_P\_SoC[ , , i\_cycle]  
# m\_M\_Exp[i\_cycle + 1, ] <- m\_M\_Exp[i\_cycle, ] %\*% m\_P\_Exp[ , , i\_cycle]  
# }  
  
# The way this matrix multiplication works is that the value in box 1 for m\_M\_SoC is multiplied by the value in box 1 for m\_P\_SoC and so on. If box 1 for m\_M\_SoC and m\_P\_SoC is the 1 (i.e. 100% of people in the PFS state when this all starts), multiplied by the PFS state probabilities, i.e., the probability of going from the PFS state into other states like so:  
  
# PFS AE1 AE2 AE3 OS Dead  
# PFS 0.974925 0.0194985 0.0194985 0.0194985 0.005074995 0.020000000  
  
# Then things will multiply correctly and we're multiplying PFS transition probabilities by the 100% of the cohort in the PFS state.   
  
# Even, if ordering v\_names\_states <- c("PFS", "AE1", "AE2", "AE3", "OS", "Dead") is not in the order as above, i.e., v\_names\_states <- c("OS", "AE1", "AE2", "AE3", "PFS", "Dead") , then the m\_M\_SoC will have 0, 0, 0, 0, 1, 0, because everyone still starts in the PFS state, and m\_P\_SoC will have "OS", "AE1", "AE2", "AE3", "PFS", "Dead", with the right probabilities put in the right spots, so when you matrix multiply it things will multiply out fine.  
  
# However, this is not the case for ordering costs and utilities:   
  
  
# v\_tc\_SoC <- m\_M\_SoC %\*% c(c\_F\_SoC, c\_AE1, c\_AE2, c\_AE3, c\_P, c\_D)  
# v\_tc\_Exp <- m\_M\_Exp %\*% c(c\_F\_Exp, c\_AE1, c\_AE2, c\_AE3, c\_P, c\_D)  
  
  
# v\_tu\_SoC <- m\_M\_SoC %\*% c(u\_F, u\_AE1, u\_AE2, u\_AE3, u\_P, u\_D)  
# v\_tu\_Exp <- m\_M\_Exp %\*% c(u\_F, u\_AE1, u\_AE2, u\_AE3, u\_P, u\_D)  
  
# As you can see, in both cases the ordering is set manually by how we enter things in the concatenated brackets, so in the case above where ordering v\_names\_states <- c("PFS", "AE1", "AE2", "AE3", "OS", "Dead") is ordered differently, i.e., v\_names\_states <- c("OS", "AE1", "AE2", "AE3", "PFS", "Dead") when we matrix multiply costs and utilities by m\_M\_SoC and m\_M\_Exp above we will be multiplying the utility of being in the progression free state u\_F by the matrix of individuals in the OS state, which will clearly be a smaller number of individuals in the first few waves, and multiplying the utility of the OS state (u\_P) by the larger number of individuals actually in the PFS state <- c("OS", "AE1", "AE2", "AE3", "PFS", "Dead"). We'll be doing the same thing with costs. What this will mean is more people getting the OS costs and OS utility and fewer people getting the PFS costs and the PFS utility, which will in turn have consequences for the cost-effectiveness analysis results with more OS costs and more OS utility being considered in the equation that compares costs and utilities.  
  
# I've confirmed all of the things I say about by changing the ordering first of v\_names\_states, and then of the cost and utility concatenations. Changing the ordering of v\_names\_states did nothing to the CEA results or Tornado diagram provided I changed the ordering of utilities and costs to match this, changing the ordering of utilities and costs changed both unless I changed them to be in an order that matched the changed ordering of v\_names\_states.  
  
  
  
  
  
  
# At the bottom of the parametric survival code I think about how time from individual patient data may be changed to match the time of our cycles.  
  
# I think the digitiser will give me the time in the context of the survival curves I am digitising, i.e., time in weeks, or time in months or time in years.  
  
# Then I will have to set-up my time accordingly in the R code so that my cycle length is at the same level as the individual patient data.  
  
# That is, in Koen's example:  
  
# C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\ISPOR WEBINAR Health Economic Modelling in R\ISPOR\_webinar\_R-master  
  
# the data includes a column for the time (in years)  
# t\_cycle <- 1/4 # cycle length of 3 months (in years) # n\_cycle <- 60 # number of cycles (total runtime 15 years)  
   
  
# So I would have my colum for time, in the [TIME] the graph I was digitising used.  
# Then I would create my cycle length of X weeks (in [TIME] the graph I was digitising used)  
# Then I would have my number of cycles that would add up to give me a total runtime of how long I want to run the model for.  
# So, above Koen wanted to run the model for 15 years, but his cycles were in 3 months, or each cycle was a quarter of a year, so 60 quarters, or 60 3 month cycles, is 15 years.  
  
# REALISE HERE THEAT P\_PD ISNT THE PROBABILITY OF PROGRESSION TO DEAD, BUT OF PFS TO DEAD, OF FIRST LINE TO DEAD, BECAUSE OUR APD CURVES ONLY EVER DESCRIBE FIRST LINE TREATMENT, BE THAT FIRST LINE SOC TREATMENT OR FIRST LINE EXP TREATMENT.  
  
# Here we define all our model parameters, so that we can call on these parameters later during our model:  
  
t\_cycle <- 14 # cycle length of 2 weeks (in [[days]] - this is assuming the survival curves I am digitising will be in [[days]] if they are in another period I will have to represent my cycle length in that period instead).   
n\_cycle <- 143   
# We set the number of cycles to 143 to reflect 2,000 days from the Angiopredict study (5 Years, 5 Months, 3 Weeks, 1 Day) broken down into fortnightly cycles  
v\_names\_cycles <- paste("cycle", 0:n\_cycle)   
# So here, we just name each cycle by the cycle its on, going from 0 up to the number of cycles there are, here 67  
v\_names\_states <- c("PFS", "OS", "Dead")   
# These are the health states in our model, PFS, Adverse Event 1, Adverse Event 2, Adverse Event 3, OS, Death.  
n\_states <- length(v\_names\_states)   
# We're just taking the number of health states from the number of names we came up with, i.e. the number of names to reflect the number of health states   
  
# Strategy names  
v\_names\_strats <- c("Standard of Care",   
 "Experimental Treatment")  
 # store the strategy names  
n\_str <- length(v\_names\_strats)   
# number of strategies  
  
  
  
# TRANSITION PROBABILITIES: Time-To-Transition - TTP:  
  
  
# Time-dependent transition probabilities are obtained in four steps  
# 1) Defining the cycle times  
# 2) Obtaining the event-free (i.e. survival) probabilities for the cycle times for SoC  
# 3) Obtaining the event-free (i.e. survival) probabilities for the cycle times for Exp based on a hazard ratio  
# 4) Obtaining the time-dependent transition probabilities from the event-free (i.e. survival) probabilities  
  
# 1) Defining the cycle times  
(t <- seq(from = 0, by = t\_cycle, length.out = n\_cycle + 1))

## [1] 0 14 28 42 56 70 84 98 112 126 140 154 168 182 196  
## [16] 210 224 238 252 266 280 294 308 322 336 350 364 378 392 406  
## [31] 420 434 448 462 476 490 504 518 532 546 560 574 588 602 616  
## [46] 630 644 658 672 686 700 714 728 742 756 770 784 798 812 826  
## [61] 840 854 868 882 896 910 924 938 952 966 980 994 1008 1022 1036  
## [76] 1050 1064 1078 1092 1106 1120 1134 1148 1162 1176 1190 1204 1218 1232 1246  
## [91] 1260 1274 1288 1302 1316 1330 1344 1358 1372 1386 1400 1414 1428 1442 1456  
## [106] 1470 1484 1498 1512 1526 1540 1554 1568 1582 1596 1610 1624 1638 1652 1666  
## [121] 1680 1694 1708 1722 1736 1750 1764 1778 1792 1806 1820 1834 1848 1862 1876  
## [136] 1890 1904 1918 1932 1946 1960 1974 1988 2002

# I think here we're saying, at each cycle how many of the time periods our individual patient data is measured at have passed? Here our individual patient data is in days, so we have 0 in cycle 0, 14 (or two weeks) in cycle 1, and so on.  
  
# Having established that allows us to obtain the transition probabilities for the time we are interested in for our cycles from this different period individual patient data, so where the individual patient data is in days and our cycles are in fortnight or half months, this allows us to obtain transition probabilities for these fortnights.  
  
# 2) Obtaining the event-free (i.e. survival) probabilities for the cycle times for SoC  
# S\_FP\_SoC - survival of progression free to progression, i.e. not going to progression, i.e. staying in progression free.  
# Note that the coefficients [that we took from flexsurvreg earlier] need to be transformed to obtain the parameters that the base R function uses  
  
  
S\_FP\_SoC <- pweibull(  
 q = t,   
 shape = exp(coef\_weibull\_shape\_SoC),   
 scale = exp(coef\_weibull\_scale\_SoC),   
 lower.tail = FALSE  
)  
  
head(cbind(t, S\_FP\_SoC))

## t S\_FP\_SoC  
## [1,] 0 1.0000000  
## [2,] 14 0.9851528  
## [3,] 28 0.9615106  
## [4,] 42 0.9333186  
## [5,] 56 0.9021389  
## [6,] 70 0.8689305

# t S\_FP\_SoC  
# [1,] 0.0 1.0000000  
# [2,] 0.5 0.9948214  
# [3,] 1.0 0.9770661  
# [4,] 1.5 0.9458256  
# [5,] 2.0 0.9015175  
# [6,] 2.5 0.8454597  
  
  
# Having the above header shows that this is probability for surviving in the F->P state, i.e., staying in this state, because you can see in time 0 100% of people are in this state, meaning 100% of people hadnt progressed and were in PFS, if this was instead about the progressed state (i.e. OS), there should be no-one in this state when the model starts, as everyone starts in the PFS state, and it takes a while for people to reach the OS state.  
  
  
# 3) Obtaining the event-free (i.e. survival) probabilities for the cycle times for Experimental treatment (aka the novel therapy) based on a hazard ratio.  
# So here we basically have a hazard ratio for the novel therapy that says you do X much better under the novel therapy than under standard of care, and we want to apply it to standard of care from our individual patient data to see how much improved things would be under the novel therapy.  
# (NB - if we ultimately decide not to use a hazard ratio, I could probably just create my transition probabilities for the experimental therapy from individual patient data that I have digitised from patients under this novel therapy).  
# Here our hazard ratio is 0.6, I can change that for our hazard ratio.  
# - note that S(t) = exp(-H(t)) and, hence, H(t) = -ln(S(t))  
# that is, the survival function is the expoential of the negative hazard function, per:  
# https://faculty.washington.edu/yenchic/18W\_425/Lec5\_survival.pdf  
# and:   
# https://web.stanford.edu/~lutian/coursepdf/unit1.pdf  
# Also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\flexsurv  
# And to multiply by the hazard ratio it's necessary to convert the survivor function into the hazard function, multiply by the hazard ratio, and then convert back to the survivor function, and then these survivor functions are used for the probabilities.  
HR\_FP\_Exp <- 0.68  
H\_FP\_SoC <- -log(S\_FP\_SoC)  
H\_FP\_Exp <- H\_FP\_SoC \* HR\_FP\_Exp  
S\_FP\_Exp <- exp(-H\_FP\_Exp)  
  
head(cbind(t, S\_FP\_SoC, H\_FP\_SoC, H\_FP\_Exp, S\_FP\_Exp))

## t S\_FP\_SoC H\_FP\_SoC H\_FP\_Exp S\_FP\_Exp  
## [1,] 0 1.0000000 0.00000000 0.00000000 1.0000000  
## [2,] 14 0.9851528 0.01495856 0.01017182 0.9898797  
## [3,] 28 0.9615106 0.03924965 0.02668976 0.9736633  
## [4,] 42 0.9333186 0.06900863 0.04692587 0.9541581  
## [5,] 56 0.9021389 0.10298683 0.07003104 0.9323649  
## [6,] 70 0.8689305 0.14049218 0.09553468 0.9088868

head(cbind(t, S\_FP\_SoC, H\_FP\_SoC))

## t S\_FP\_SoC H\_FP\_SoC  
## [1,] 0 1.0000000 0.00000000  
## [2,] 14 0.9851528 0.01495856  
## [3,] 28 0.9615106 0.03924965  
## [4,] 42 0.9333186 0.06900863  
## [5,] 56 0.9021389 0.10298683  
## [6,] 70 0.8689305 0.14049218

# I want to vary my probabilities for the one-way sensitivity analysis, particularly for the tornado plot of the deterministic sensitivity analysis.   
   
 # The problem here is that df\_params\_OWSA doesnt like the fact that a different probability for each cycle (from the time-dependent transition probabilities) gives 122 rows (because there are 60 cycles, two treatment strategies and a probability for each cycle). It wants the same number of rows as there are probabilities, i.e., it would prefer a probability of say 0.50 and then a max and a min around that.  
   
 # To address this, I think I can apply this mean, max and min to the hazard ratios instead, knowing that when run\_owsa\_det is run in the sensitivity analysis it calls this function to run and in this function the hazard ratios generate the survivor function, and then these survivor functions are used to generate the probabilities (which will be cycle dependent).  
   
 # This is fine for the hazard ratio for the experimental strategy, I can just take:  
   
 # HR\_FP\_Exp as my mean, and:  
   
 # Minimum\_HR\_FP\_Exp <- HR\_FP\_Exp - 0.20\*HR\_FP\_Exp  
 # Maximum\_HR\_FP\_Exp <- HR\_FP\_Exp + 0.20\*HR\_FP\_Exp  
   
 # For min and max.  
   
 # For standard of care there was no hazard ratio, because we took these values from the survival curves directly, and didnt vary them by a hazard ratio, like we do above.  
   
 # To address this, I create a hazard ratio that is exactly one.  
   
 # hazard ratio  
  
 # A measure of how often a particular event happens in one group compared to how often it happens in another group, over time. In cancer research, hazard ratios are often used in clinical trials to measure survival at any point in time in a group of patients who have been given a specific treatment compared to a control group given another treatment or a placebo. A hazard ratio of one means that there is no difference in survival between the two groups. A hazard ratio of greater than one or less than one means that survival was better in one of the groups. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/hazard-ratio  
  
 # Thus, I can have a hazard ratio where the baseline value of it gives you the survival curves, and thus the probabilities, from the actual survival curves we are drawing from, and where the min and max will be 1 +/- 0.20, which will give us probabilities that are 20% higher or lower than the probabilities from the actual survival curves that we are drawing from in the parametric survival analysis to get transitions under standard of care.  
   
 # To do this, I just have to add a hazard ratio to the code that creates the transition probabilities under standard of care as below, then I can add that hazard ratio, and it's max and min, to the deterministic sensitivity analysis and vary all the probabilities by 20%.  
   
  
 # So here we basically have a hazard ratio that is equal to 1, so it leaves things unchanged for patients, and we want to apply it to standard of care from our individual patient data to leave things unchanged in this function, but allow things to change in the sensitivity analysis.  
   
 # Here our hazard ratio is 1, things are unchanged.  
  
# So, first we create our hazard ratio == 1  
HR\_FP\_SoC <- 1  
  
# (I'm creating the below as new parameters, i.e. putting "nu" infront of them, in case keeping the name the same causes a problem for when I want to use them in the deterministic sensivity analysis, i.e., if I generate a parameter from itself - say var\_name = var\_name exactly, then there may be some way in which R handles code that won't let this work, or will take one parameter before the other, or something and stop the model from executing correctly).  
  
# Then, we create our hazard function for SoC:  
NU\_S\_FP\_SoC <- S\_FP\_SoC  
NU\_H\_FP\_SoC <- -log(NU\_S\_FP\_SoC)  
# Then, we multiply this hazard function by our hazard ratio, which is just 1, but which gives us the opportunity to apply a hazard ratio to standard of care in our code and thus to have a hazard ratio for standard of care for our one way deterministic sensitivity analysis and tornado diagram.  
NUnu\_H\_FP\_SoC <- NU\_H\_FP\_SoC \* HR\_FP\_SoC  
# Again, I was worried that with overlap when creating parameters I would have a problem with the deterministic sensivity analysis so I call it NU again to make it a "new" parameter again.  
NU\_S\_FP\_SoC <- exp(-NUnu\_H\_FP\_SoC)  
  
head(cbind(t, NU\_S\_FP\_SoC, NUnu\_H\_FP\_SoC))

## t NU\_S\_FP\_SoC NUnu\_H\_FP\_SoC  
## [1,] 0 1.0000000 0.00000000  
## [2,] 14 0.9851528 0.01495856  
## [3,] 28 0.9615106 0.03924965  
## [4,] 42 0.9333186 0.06900863  
## [5,] 56 0.9021389 0.10298683  
## [6,] 70 0.8689305 0.14049218

# NU\_H\_FP\_SoC <- -log(NU\_S\_FP\_SoC)  
# # Then, we multiply this hazard function by our hazard ratio, which is just 1, but which gives us the opportunity to apply a hazard ratio to standard of care in our code and thus to have a hazard ratio for standard of care for our one way deterministic sensitivity analysis and tornado diagram.  
# NU\_H\_FP\_SoC <- NU\_H\_FP\_SoC \* HR\_FP\_SoC  
# #   
# NU\_S\_FP\_SoC <- exp(-NU\_H\_FP\_SoC)  
#   
# head(cbind(t, NU\_S\_FP\_SoC, NU\_H\_FP\_SoC))  
  
  
  
  
  
  
  
  
  
  
  
  
# 4) Obtaining the time-dependent transition probabilities from the event-free (i.e. survival) probabilities  
  
# Now we can take the probability of being in the PFS state at each of our cycles, as created above, from 100% (i.e. from 1) in order to get the probability of NOT being in the PFS state, i.e. in order to get the probability of moving into the progressed state, or the OS state.  
   
p\_PFSOS\_SoC <- p\_PFSOS\_Exp <- rep(NA, n\_cycle)  
  
# First we make the probability of going from progression-free (F) to progression (P) blank (i.e. NA) for all the cycles in standard of care and all the cycles under the experimental strategy.  
  
for(i in 1:n\_cycle) {  
 p\_PFSOS\_SoC[i] <- 1 - NU\_S\_FP\_SoC[i+1] / NU\_S\_FP\_SoC[i]  
 p\_PFSOS\_Exp[i] <- 1 - S\_FP\_Exp[i+1] / S\_FP\_Exp[i]  
}  
  
  
  
  
# If I ever wanted to round my probabilities to 2 decimal places, I can do this as below, but Koen's code actually makes probabilities that sum perfectly to 1, so there's no longer any need to do this.  
  
# round(p\_PFSOS\_SoC, digits=2)  
# round(p\_PFSOS\_Exp, digits=2)  
  
# Then we generate our transition probability under standard of care and under the experimental treatement using survival functions that havent and have had the hazard ratio from above applied to them, respectively.  
  
  
# The way this works is the below, you take next cycles probability of staying in this state, divide it by this cycles probability of staying in this state, and take it from 1 to get the probability of leaving this state.   
  
# > head(cbind(t, S\_FP\_SoC))  
# t S\_FP\_SoC  
# [1,] 0.0 1.0000000  
# [2,] 0.5 0.9948214  
# [3,] 1.0 0.9770661  
# [4,] 1.5 0.9458256  
# [5,] 2.0 0.9015175  
# [6,] 2.5 0.8454597  
# > 1-0.9948214/1.0000000  
# [1] 0.0051786  
# > 0.9770661/0.9948214  
# [1] 0.9821523  
# > 1-0.9821523  
# [1] 0.0178477  
  
# p\_FP\_SoC  
p\_PFSOS\_SoC

## [1] 0.01484724 0.02399843 0.02932054 0.03340742 0.03681073 0.03976670  
## [7] 0.04240199 0.04479366 0.04699259 0.04903438 0.05094508 0.05274438  
## [13] 0.05444758 0.05606686 0.05761207 0.05909131 0.06051136 0.06187791  
## [19] 0.06319584 0.06446934 0.06570203 0.06689710 0.06805734 0.06918523  
## [25] 0.07028297 0.07135252 0.07239566 0.07341399 0.07440894 0.07538184  
## [31] 0.07633389 0.07726618 0.07817972 0.07907543 0.07995417 0.08081674  
## [37] 0.08166385 0.08249621 0.08331443 0.08411910 0.08491079 0.08569001  
## [43] 0.08645723 0.08721292 0.08795750 0.08869138 0.08941493 0.09012851  
## [49] 0.09083246 0.09152710 0.09221274 0.09288966 0.09355813 0.09421841  
## [55] 0.09487076 0.09551541 0.09615257 0.09678247 0.09740531 0.09802128  
## [61] 0.09863057 0.09923336 0.09982982 0.10042011 0.10100440 0.10158282  
## [67] 0.10215553 0.10272267 0.10328436 0.10384074 0.10439193 0.10493805  
## [73] 0.10547921 0.10601553 0.10654711 0.10707406 0.10759647 0.10811444  
## [79] 0.10862806 0.10913743 0.10964263 0.11014374 0.11064084 0.11113402  
## [85] 0.11162334 0.11210889 0.11259074 0.11306895 0.11354359 0.11401473  
## [91] 0.11448243 0.11494675 0.11540776 0.11586550 0.11632004 0.11677144  
## [97] 0.11721974 0.11766499 0.11810726 0.11854658 0.11898301 0.11941659  
## [103] 0.11984737 0.12027539 0.12070069 0.12112332 0.12154333 0.12196074  
## [109] 0.12237559 0.12278794 0.12319781 0.12360524 0.12401026 0.12441291  
## [115] 0.12481323 0.12521125 0.12560699 0.12600050 0.12639180 0.12678092  
## [121] 0.12716789 0.12755275 0.12793551 0.12831621 0.12869488 0.12907153  
## [127] 0.12944621 0.12981893 0.13018971 0.13055859 0.13092558 0.13129071  
## [133] 0.13165400 0.13201548 0.13237517 0.13273309 0.13308925 0.13344369  
## [139] 0.13379642 0.13414746 0.13449683 0.13484455 0.13519064

#> p\_FP\_SoC  
# [1] 0.005178566 0.017847796 0.031973721 0.046845943 0.062181645  
#p\_FP\_Exp  
p\_PFSOS\_Exp

## [1] 0.01012026 0.01638227 0.02003273 0.02284029 0.02518117 0.02721649  
## [7] 0.02903271 0.03068241 0.03220033 0.03361079 0.03493156 0.03617610  
## [13] 0.03735488 0.03847620 0.03954680 0.04057223 0.04155710 0.04250533  
## [19] 0.04342023 0.04430468 0.04516116 0.04599185 0.04679865 0.04758326  
## [25] 0.04834719 0.04909178 0.04981825 0.05052769 0.05122109 0.05189935  
## [31] 0.05256329 0.05321367 0.05385117 0.05447643 0.05509003 0.05569251  
## [37] 0.05628438 0.05686611 0.05743812 0.05800083 0.05855461 0.05909981  
## [43] 0.05963677 0.06016579 0.06068718 0.06120120 0.06170812 0.06220818  
## [49] 0.06270162 0.06318865 0.06366949 0.06414432 0.06461334 0.06507673  
## [55] 0.06553465 0.06598726 0.06643473 0.06687719 0.06731480 0.06774767  
## [61] 0.06817594 0.06859973 0.06901916 0.06943434 0.06984538 0.07025239  
## [67] 0.07065545 0.07105467 0.07145014 0.07184195 0.07223018 0.07261492  
## [73] 0.07299624 0.07337421 0.07374892 0.07412043 0.07448882 0.07485414  
## [79] 0.07521646 0.07557585 0.07593235 0.07628604 0.07663697 0.07698518  
## [85] 0.07733073 0.07767368 0.07801408 0.07835196 0.07868738 0.07902038  
## [91] 0.07935100 0.07967930 0.08000530 0.08032905 0.08065059 0.08096995  
## [97] 0.08128718 0.08160230 0.08191536 0.08222638 0.08253541 0.08284246  
## [103] 0.08314758 0.08345079 0.08375213 0.08405162 0.08434929 0.08464517  
## [109] 0.08493929 0.08523166 0.08552233 0.08581130 0.08609862 0.08638429  
## [115] 0.08666835 0.08695082 0.08723172 0.08751107 0.08778889 0.08806520  
## [121] 0.08834003 0.08861339 0.08888531 0.08915580 0.08942488 0.08969256  
## [127] 0.08995888 0.09022384 0.09048747 0.09074977 0.09101077 0.09127048  
## [133] 0.09152892 0.09178610 0.09204204 0.09229675 0.09255025 0.09280256  
## [139] 0.09305368 0.09330363 0.09355242 0.09380008 0.09404660

# TRANSITION PROBABILITIES: Time-To-Dead TTD  
  
# REALISE HERE THEAT P\_PD ISNT THE PROBABILITY OF PROGRESSION TO DEAD, BUT OF PFS TO DEAD, OF FIRST LINE TO DEAD, BECAUSE OUR APD CURVES ONLY EVER DESCRIBE FIRST LINE TREATMENT, BE THAT FIRST LINE SOC TREATMENT OR FIRST LINE EXP TREATMENT.  
  
  
# To make sure that my PFS probabilities only reflect going from PFS to progression, I create the probability of going from PFS to DEAD under standard of care and the experimental, and decrease my PFS to progression probability by the probability of going into the dead state, such that I am only capturing people going into progression, and not people going into death as well.   
  
  
# So, first I create the transition probabilities of progression free into dead for SoC and Exp, then I convert all the probabilities (i.e. those for PFS and those for OS) into rates, minus them from eachother, turn them back into probabilities, and make sure none are negative (and where they are replace these with 0).  
  
# Actually, I don't do the rates thing, I just I take all the probabilities (i.e. those for PFS and those for OS), minus them from eachother.  
  
  
# Time-dependent transition probabilities are obtained in four steps  
# 1) Defining the cycle times [we already did this above]  
# 2) Obtaining the event-free (i.e. overall survival) probabilities for the cycle times for SoC  
# 3) Obtaining the event-free (i.e. overall survival) probabilities for the cycle times for Exp based on a hazard ratio if we think we will be applying a hazard ratio in the OS -> Death setting. Probably not, probably what we'll be doing is saying that once you get into the OS state under the experimental strategy, you recieve the same second-line treatment as standard of care again and thus your event-free (i.e. overall survival) probabilities for the cycle times are the same as for SoC. - Actually, I create transition probabilities from treatment to death under SoC and the Exp treatment here such that I can take them from the transition probabilities from treatment to progression for SoC and Exp treatment, because the OS here from Angiopredict is transitioning from the first line treatment to dead, not from second line treatment to death, and once we get rid of the people who were leaving first line treatment to die in PFS, all we have left is people leaving first line treatment to progress. And then we can keep the first line treatment to death probabilities we've created from the OS curves to capture people who have left first line treatment to transition into death rather than second line treatment.  
# 4) Obtaining the time-dependent transition probabilities from the event-free (i.e. overall survival) probabilities  
  
# 1) Defining the cycle times  
(t <- seq(from = 0, by = t\_cycle, length.out = n\_cycle + 1))

## [1] 0 14 28 42 56 70 84 98 112 126 140 154 168 182 196  
## [16] 210 224 238 252 266 280 294 308 322 336 350 364 378 392 406  
## [31] 420 434 448 462 476 490 504 518 532 546 560 574 588 602 616  
## [46] 630 644 658 672 686 700 714 728 742 756 770 784 798 812 826  
## [61] 840 854 868 882 896 910 924 938 952 966 980 994 1008 1022 1036  
## [76] 1050 1064 1078 1092 1106 1120 1134 1148 1162 1176 1190 1204 1218 1232 1246  
## [91] 1260 1274 1288 1302 1316 1330 1344 1358 1372 1386 1400 1414 1428 1442 1456  
## [106] 1470 1484 1498 1512 1526 1540 1554 1568 1582 1596 1610 1624 1638 1652 1666  
## [121] 1680 1694 1708 1722 1736 1750 1764 1778 1792 1806 1820 1834 1848 1862 1876  
## [136] 1890 1904 1918 1932 1946 1960 1974 1988 2002

# 2) Obtaining the event-free (i.e. overall survival) probabilities for the cycle times for SoC  
# S\_PD\_SoC - survival of progression to dead, i.e. not going to dead, i.e. staying in progression.  
# Note that the coefficients [that we took from flexsurvreg earlier] need to be transformed to obtain the parameters that the base R function uses  
  
  
S\_PD\_SoC <- pweibull(  
 q = t,   
 shape = exp(coef\_TTD\_weibull\_shape\_SoC),   
 scale = exp(coef\_TTD\_weibull\_scale\_SoC),   
 lower.tail = FALSE  
)  
  
head(cbind(t, S\_PD\_SoC))

## t S\_PD\_SoC  
## [1,] 0 1.0000000  
## [2,] 14 0.9952916  
## [3,] 28 0.9880164  
## [4,] 42 0.9793493  
## [5,] 56 0.9696727  
## [6,] 70 0.9592044

# Having the above header shows that this is probability for surviving in the P->D state, i.e., staying in this state, because you should see in time 0 0% of people are in this state, meaning 100% of people hadnt gone into the progressed state and were in PFS, which make sense in this model, the model starts with everyone in PFS, no-one starts the model in OS, and it takes a while for people to reach the OS state.  
  
  
# 3) Obtaining the event-free (i.e. overall survival) probabilities for the cycle times for Experimental treatment (aka the novel therapy) based on a hazard ratio.  
# So here we basically have a hazard ratio for the novel therapy that says you do X much better under the novel therapy than under standard of care, and we want to apply it to standard of care from our individual patient data to see how much improved things would be under the novel therapy.  
  
# Here our hazard ratio is 0.6, I can change that for our hazard ratio.  
# - note that S(t) = exp(-H(t)) and, hence, H(t) = -ln(S(t))  
# that is, the survival function is the expoential of the negative hazard function, per:  
# https://faculty.washington.edu/yenchic/18W\_425/Lec5\_survival.pdf  
# and:   
# https://web.stanford.edu/~lutian/coursepdf/unit1.pdf  
# Also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\flexsurv  
# And to multiply by the hazard ratio it's necessary to convert the survivor function into the hazard function, multiply by the hazard ratio, and then convery back to the survivor function, and then these survivor functions are used for the probabilities.  
HR\_PD\_Exp <- 0.65  
H\_PD\_SoC <- -log(S\_PD\_SoC)  
H\_PD\_Exp <- H\_PD\_SoC \* HR\_PD\_Exp  
S\_PD\_Exp <- exp(-H\_PD\_Exp)  
  
head(cbind(t, S\_PD\_SoC, H\_PD\_SoC, H\_PD\_Exp, S\_PD\_Exp))

## t S\_PD\_SoC H\_PD\_SoC H\_PD\_Exp S\_PD\_Exp  
## [1,] 0 1.0000000 0.000000000 0.000000000 1.0000000  
## [2,] 14 0.9952916 0.004719564 0.003067716 0.9969370  
## [3,] 28 0.9880164 0.012055994 0.007836396 0.9921942  
## [4,] 42 0.9793493 0.020866948 0.013563516 0.9865281  
## [5,] 56 0.9696727 0.030796701 0.020017856 0.9801812  
## [6,] 70 0.9592044 0.041651043 0.027073178 0.9732900

# If I decide that, as I said, once you get into the OS state under the experimental strategy, you recieve the same second-line treatment as standard of care again and thus your event-free (i.e. overall survival) probabilities for the cycle times are the same as for SoC, then I can use the following coding - which is just repeating what I did for standard of care but this time giving it to the experimental stratgey:  
#   
# S\_PD\_Exp <- pweibull(  
# q = t,   
# shape = exp(coef\_TTD\_weibull\_shape\_SoC),   
# scale = exp(coef\_TTD\_weibull\_scale\_SoC),   
# lower.tail = FALSE  
# )  
#   
# head(cbind(t, S\_PD\_Exp))  
  
# I've coded in both options here and I can make a decision when applying this.  
  
  
  
  
# I want to vary my probabilities for the one-way sensitivity analysis, particularly for the tornado plot of the deterministic sensitivity analysis.   
  
# The problem here is that df\_params\_OWSA doesnt like the fact that a different probability for each cycle (from the time-dependent transition probabilities) gives 122 rows (because there are 60 cycles, two treatment strategies and a probability for each cycle). It wants the same number of rows as there are probabilities, i.e., it would prefer a probability of say 0.50 and then a max and a min around that.  
  
# To address this, I think I can apply this mean, max and min to the hazard ratios instead, knowing that when run\_owsa\_det is run in the sensitivity analysis it calls this function to run and in this function the hazard ratios generate the survivor function, and then these survivor functions are used to generate the probabilities (which will be cycle dependent).  
  
# This is fine for the hazard ratio for the experimental strategy, I can just take:  
  
# HR\_PD\_Exp as my mean, and:  
  
# Minimum\_HR\_PD\_Exp <- HR\_PD\_Exp - 0.20\*HR\_PD\_Exp  
# Maximum\_HR\_PD\_Exp <- HR\_PD\_Exp + 0.20\*HR\_PD\_Exp  
  
# For min and max.  
  
# For standard of care there was no hazard ratio, because we took these values from the survival curves directly, and didnt vary them by a hazard ratio, like we do above.  
  
# To address this, I create a hazard ratio that is exactly one.  
  
# hazard ratio  
  
# A measure of how often a particular event happens in one group compared to how often it happens in another group, over time. In cancer research, hazard ratios are often used in clinical trials to measure survival at any point in time in a group of patients who have been given a specific treatment compared to a control group given another treatment or a placebo. A hazard ratio of one means that there is no difference in survival between the two groups. A hazard ratio of greater than one or less than one means that survival was better in one of the groups. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/hazard-ratio  
  
# Thus, I can have a hazard ratio where the baseline value of it gives you the survival curves, and thus the probabilities, from the actual survival curves we are drawing from, and where the min and max will be 1 +/- 0.20, which will give us probabilities that are 20% higher or lower than the probabilities from the actual survival curves that we are drawing from in the parametric survival analysis to get transitions under standard of care.  
  
# To do this, I just have to add a hazard ratio to the code that creates the transition probabilities under standard of care as below, then I can add that hazard ratio, and it's max and min, to the deterministic sensitivity analysis and vary all the probabilities by 20%.  
  
  
# So here we basically have a hazard ratio that is equal to 1, so it leaves things unchanged for patients, and we want to apply it to standard of care from our individual patient data to leave things unchanged in this function, but allow things to change in the sensitivity analysis.  
  
# Here our hazard ratio is 1, things are unchanged.  
  
# So, first we create our hazard ratio == 1  
HR\_PD\_SoC <- 1  
  
# (I'm creating the below as new parameters, i.e. putting "nu" infront of them, in case keeping the name the same causes a problem for when I want to use them in the deterministic sensivity analysis, i.e., if I generate a parameter from itself - say var\_name = var\_name exactly, then there may be some way in which R handles code that won't let this work, or will take one parameter before the other, or something and stop the model from executing correctly).  
  
# Then, we create our hazard function for SoC:  
NU\_S\_PD\_SoC <- S\_PD\_SoC  
NU\_H\_PD\_SoC <- -log(NU\_S\_PD\_SoC)  
# Then, we multiply this hazard function by our hazard ratio, which is just 1, but which gives us the opportunity to apply a hazard ratio to standard of care in our code and thus to have a hazard ratio for standard of care for our one way deterministic sensitivity analysis and tornado diagram.  
NUnu\_H\_PD\_SoC <- NU\_H\_PD\_SoC \* HR\_PD\_SoC  
# Again, I was worried that with overlap when creating parameters I would have a problem with the deterministic sensivity analysis so I call it NU again to make it a "new" parameter again.  
NU\_S\_PD\_SoC <- exp(-NUnu\_H\_PD\_SoC)  
  
head(cbind(t, NU\_S\_PD\_SoC, NUnu\_H\_PD\_SoC))

## t NU\_S\_PD\_SoC NUnu\_H\_PD\_SoC  
## [1,] 0 1.0000000 0.000000000  
## [2,] 14 0.9952916 0.004719564  
## [3,] 28 0.9880164 0.012055994  
## [4,] 42 0.9793493 0.020866948  
## [5,] 56 0.9696727 0.030796701  
## [6,] 70 0.9592044 0.041651043

# NU\_H\_PD\_SoC <- -log(NU\_S\_PD\_SoC)  
# # Then, we multiply this hazard function by our hazard ratio, which is just 1, but which gives us the opportunity to apply a hazard ratio to standard of care in our code and thus to have a hazard ratio for standard of care for our one way deterministic sensitivity analysis and tornado diagram.  
# NU\_H\_PD\_SoC <- NU\_H\_PD\_SoC \* HR\_PD\_SoC  
# #   
# NU\_S\_PD\_SoC <- exp(-NU\_H\_PD\_SoC)  
#   
# head(cbind(t, NU\_S\_PD\_SoC, NU\_H\_PD\_SoC))  
  
  
# 4) Obtaining the time-dependent transition probabilities from the event-free (i.e. survival) probabilities  
  
# Now we can take the probability of being in the PFS state at each of our cycles, as created above, from 100% (i.e. from 1) in order to get the probability of NOT being in the PFS state, i.e. in order to get the probability of moving into the progressed state, or the OS state.  
  
  
p\_PFSD\_SoC <- p\_PFSD\_Exp <- rep(NA, n\_cycle)  
  
# First we make the probability of going from progression-free (F) to progression (P) blank (i.e. NA) for all the cycles in standard of care and all the cycles under the experimental strategy.  
  
for(i in 1:n\_cycle) {  
 p\_PFSD\_SoC[i] <- 1 - NU\_S\_PD\_SoC[i+1] / NU\_S\_PD\_SoC[i]  
 p\_PFSD\_Exp[i] <- 1 - S\_PD\_Exp[i+1] / S\_PD\_Exp[i]  
}  
  
# round(p\_PFSD\_SoC, digits=2)  
# round(p\_PFSD\_Exp, digits=2)  
  
  
# Then we generate our transition probability under standard of care and under the experimental treatement using survival functions that havent and have had the hazard ratio from above applied to them, respectively. [If we decide not to apply a hazard ratio for the experimental strategy going from progression to dead then neither may have a hazard ratio applied to them].  
  
  
# The way this works is, you take next cycles probability of staying in this state, divide it by this cycles probability of staying in this state, and take it from 1 to get the probability of leaving this state.   
  
p\_PFSD\_SoC

## [1] 0.004708444 0.007309584 0.008772251 0.009880615 0.010795647 0.011585361  
## [7] 0.012285903 0.012919101 0.013499287 0.014036442 0.014537832 0.015008933  
## [13] 0.015453987 0.015876356 0.016278753 0.016663408 0.017032173 0.017386609  
## [19] 0.017728046 0.018057624 0.018376331 0.018685030 0.018984478 0.019275345  
## [25] 0.019558225 0.019833651 0.020102098 0.020363996 0.020619734 0.020869666  
## [31] 0.021114111 0.021353365 0.021587697 0.021817354 0.022042564 0.022263539  
## [37] 0.022480473 0.022693548 0.022902933 0.023108783 0.023311247 0.023510460  
## [43] 0.023706551 0.023899640 0.024089841 0.024277261 0.024461998 0.024644148  
## [49] 0.024823800 0.025001039 0.025175944 0.025348592 0.025519054 0.025687399  
## [55] 0.025853692 0.026017994 0.026180364 0.026340859 0.026499532 0.026656434  
## [61] 0.026811615 0.026965120 0.027116995 0.027267282 0.027416023 0.027563257  
## [67] 0.027709021 0.027853353 0.027996286 0.028137856 0.028278093 0.028417029  
## [73] 0.028554693 0.028691116 0.028826324 0.028960344 0.029093202 0.029224923  
## [79] 0.029355532 0.029485051 0.029613503 0.029740910 0.029867293 0.029992673  
## [85] 0.030117069 0.030240500 0.030362985 0.030484542 0.030605189 0.030724942  
## [91] 0.030843817 0.030961832 0.031079000 0.031195338 0.031310860 0.031425580  
## [97] 0.031539512 0.031652669 0.031765065 0.031876712 0.031987623 0.032097809  
## [103] 0.032207283 0.032316056 0.032424139 0.032531543 0.032638278 0.032744355  
## [109] 0.032849784 0.032954575 0.033058737 0.033162280 0.033265213 0.033367545  
## [115] 0.033469284 0.033570440 0.033671020 0.033771033 0.033870486 0.033969387  
## [121] 0.034067745 0.034165565 0.034262857 0.034359626 0.034455880 0.034551625  
## [127] 0.034646869 0.034741618 0.034835878 0.034929655 0.035022956 0.035115786  
## [133] 0.035208152 0.035300059 0.035391514 0.035482521 0.035573085 0.035663213  
## [139] 0.035752910 0.035842180 0.035931029 0.036019462 0.036107483

p\_PFSD\_Exp

## [1] 0.003063016 0.004757328 0.005710751 0.006433555 0.007030492 0.007545832  
## [7] 0.008003102 0.008416512 0.008795392 0.009146241 0.009473790 0.009781605  
## [13] 0.010072449 0.010348510 0.010611556 0.010863039 0.011104166 0.011335954  
## [19] 0.011559268 0.011774851 0.011983348 0.012185320 0.012381261 0.012571607  
## [25] 0.012756746 0.012937023 0.013112750 0.013284206 0.013451645 0.013615297  
## [31] 0.013775371 0.013932058 0.014085535 0.014235963 0.014383490 0.014528255  
## [37] 0.014670383 0.014809994 0.014947197 0.015082094 0.015214781 0.015345348  
## [43] 0.015473877 0.015600448 0.015725134 0.015848005 0.015969126 0.016088558  
## [49] 0.016206360 0.016322587 0.016437291 0.016550521 0.016662325 0.016772748  
## [55] 0.016881830 0.016989613 0.017096135 0.017201433 0.017305542 0.017408494  
## [61] 0.017510323 0.017611057 0.017710728 0.017809362 0.017906986 0.018003626  
## [67] 0.018099307 0.018194052 0.018287885 0.018380826 0.018472898 0.018564119  
## [73] 0.018654511 0.018744092 0.018832879 0.018920890 0.019008143 0.019094654  
## [79] 0.019180437 0.019265509 0.019349884 0.019433577 0.019516600 0.019598968  
## [85] 0.019680694 0.019761789 0.019842267 0.019922138 0.020001414 0.020080107  
## [91] 0.020158226 0.020235783 0.020312787 0.020389249 0.020465177 0.020540581  
## [97] 0.020615471 0.020689854 0.020763740 0.020837137 0.020910052 0.020982495  
## [103] 0.021054471 0.021125990 0.021197057 0.021267681 0.021337868 0.021407625  
## [109] 0.021476958 0.021545875 0.021614380 0.021682481 0.021750183 0.021817493  
## [115] 0.021884415 0.021950955 0.022017119 0.022082913 0.022148340 0.022213407  
## [121] 0.022278119 0.022342480 0.022406494 0.022470168 0.022533504 0.022596508  
## [127] 0.022659184 0.022721537 0.022783570 0.022845287 0.022906693 0.022967792  
## [133] 0.023028586 0.023089081 0.023149280 0.023209186 0.023268803 0.023328135  
## [139] 0.023387185 0.023445955 0.023504451 0.023562674 0.023620628

# Finally, now that I create transition probabilities from treatment to death under SoC and the Exp treatment I can take them from the transition probabilities from treatment to progression for SoC and Exp treatment, because the OS here from Angiopredict is transitioning from the first line treatment to dead, not from second line treatment to death, and once we get rid of the people who were leaving first line treatment to die in PFS, all we have left is people leaving first line treatment to progress. And then we can keep the first line treatment to death probabilities we've created from the OS curves to capture people who have left first line treatment to transition into death rather than second line treatment.  
  
# p\_PFSOS\_SoC  
# p\_PFSD\_SoC  
# p\_PFSOS\_SoC <- p\_PFSOS\_SoC - p\_PFSD\_SoC  
# p\_PFSOS\_SoC  
#   
# p\_PFSOS\_Exp  
# p\_PFSD\_Exp  
# p\_PFSOS\_Exp <- p\_PFSOS\_Exp - p\_PFSD\_Exp  
# p\_PFSOS\_Exp  
  
  
  
  
# Time-constant transition probabilities [ADVERSE EVENTS]:  
  
  
# To create transition probabilities from longer time periods I can use the information in this email to Daniel:  
  
# Inquiry re: Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer  
# - https://outlook.office.com/mail/id/AAQkAGI5OWU0NTJkLTEzMjgtNGVhOS04ZGZiLWZkOGU1MDg3ZmE5MAAQAHQCBS2m%2B%2FVAjAc%2FWSCjQEQ%3D  
  
  
# There may also be some relevant information in the below:  
  
  
## Transition probabilities and hazard ratios  
  
  
# "Note: To calculate the probability of dying from S1 and S2, use the hazard ratios provided. To do so, first convert the probability of dying from healthy, p\_HD , to a rate; then multiply this rate by the appropriate hazard ratio; finally, convert this rate back to a probability. Recall that you can convert between rates and probabilities using the following formulas: r = − log(1 − p) and p = 1 − e ( − rt ) . The package darthtools also has the functions prob\_to\_rate and rate\_to\_prob that might be of use to you." per: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_25\3\_cSTM - history dependence\_material\Download exercise handout   
  
# ?rate\_to\_prob will tell you more about this function.  
# ?prob\_to\_rate will tell you more about this function.  
  
# You can see conversions from probabilities to rates here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Model Calibration in R\UNZIP to Working Directory\_Calibration Participant Materials\ISPOR Calibration Participant Materials\SickSicker\_MarkovModel\_Function.R  
  
# As will the 50 minute mark of: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_25\Live Session Recording\Live Session Recording August 25th WITH CHAT.mkv  
  
  
# The above also describes how to convert probabilities for different time scales, i.e., convert a probability for 5 years to 1 year, etc., and how to convert data that exists as a rate to a probability for use in a Markov model.  
  
  
# The intial probabilities before I decided to do parametric survival analysis:  
  
# p\_HS\_SoC <- 0.05 # probability of becoming OS when PFS, conditional on surviving, under standard of care  
# p\_HS\_trtA <- 0.04 # probability of becoming OS when PFS, conditional on surviving, under EPI Assay  
# p\_HS\_trtB <- 0.02 # probability of becoming OS when PFS, conditional on surviving, under HDX Assay  
# p\_SD <- 0.1 # probability of dying   
# p\_HD <- 0.01 # probability of dying when PFS  
  
  
# H = HEALTHY (PFS) -> HS MEANS HEALTHY TO SICK, HD -> MEANS HEALTHY TO DEAD.  
# S = SICK (OS) -> SD MEANS SICK TO DEAD.  
# D = DEAD (DEAD)   
  
# trtA -> Means the first intervention I am studying, i.e. treatment A or the first assay.  
# trtB -> Means the second intervention I am studying, i.e. treatment B or the second assay.  
  
# To add age specific mortality to our model, we would use this #03 input model parameters of:  
  
# "C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modeling for Public Health\_DARTH\5\_Nov\_29\4\_Cohort state-transition models (cSTM) - time-dependent models\_material\Markov\_3state\_time"  
  
# with the 55 minute mark of: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_24\Live Session Recording\Live Session Recording August 24th.mp4  
  
# and this would allow us to create a vector of transition probabilities for p\_HD above, i.e., from PFS to dead, that is a little bit larger at each cycle, starting at our chosen minimum value at the first cycle and increasing each cycle until it reaches our chosen maximum value at the last cycle.  
  
# Alternatively, C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_24\4\_cSTM - time-dependent models\_material shows you how to use a life-table, as does the material from the York course, but I really think there's no need to get that detailed in our own analysis.  
  
  
  
  
# So, I now need to change p\_FD\_SoC and p\_FD\_Exp to reflect OS -> DEAD, becuase I now REALISE HERE TEAT P\_PD ISNT THE PROBABILITY OF PROGRESSION TO DEAD, BUT OF PFS TO DEAD, OF FIRST LINE TO DEAD, BECAUSE OUR APD CURVES ONLY EVER DESCRIBE FIRST LINE TREATMENT, BE THAT FIRST LINE SOC TREATMENT OR FIRST LINE EXP TREATMENT.  
 # and the FD below instead reflects the probability of going to death in the second line treatment, and I make the assumption that everyone gets the same second line treatment and give them the same probability under exp and under SoC to go from the second line therapy into dead.  
  
P\_OSD\_SoC <- 0.17 # Probability of dying when in OS.  
P\_OSD\_Exp <- 0.17 # Probability of dying when in OS.  
  
# Base-Case: 0.17 Min: 0.12 Max:0.22 According to: Wen, F., Zheng, H., Wu, Y., Wheeler, J., Zeng, X., Fu, P., & Li, Q. (2016). Cost-effectiveness Analysis of Fluorouracil, Leucovorin, and Irinotecan versus Epirubicin, Cisplatin, and Capecitabine in Patients with Advanced Gastric Adenocarcinoma. Scientific reports, 6(1), 1-8. https://sci-hub.ru/10.1038/srep36060  
  
# When digitising a progression-free survival curve everyone on that curve is on PFS, and if they leave that curve it is because their disease has progressed (so they've gone from progression free to progression, or from PFS to OS) or they've died:  
  
# "Progression-free survival (PFS) is defined as the time from random assignment in a clinical trial to disease progression or death from any cause." https://www.ncbi.nlm.nih.gov/books/NBK137763/  
  
# "Progression-free survival (PFS) is "the length of time during and after the treatment of a disease, such as cancer, that a patient lives with the disease but it does not get worse". https://en.wikipedia.org/wiki/Progression-free\_survival#:~:text=Progression-free%20survival%20(PFS),it%20does%20not%20get%20worse".  
  
# Time to Progression:   
  
# "The length of time from the date of diagnosis or the start of treatment for a disease until the disease starts to get worse or spread to other parts of the body. In a clinical trial, measuring the time to progression is one way to see how well a new treatment works. Also called TTP." https://www.cancer.gov/publications/dictionaries/cancer-terms/def/time-to-progression  
  
# Koen's data is time to progression data, so we can't figure out time to death from it. It may be possible to do this with the data I end up using, or as advised by the C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\flexsurv folder, but at the moment I'll just set PFS to Dead as a time-constant transition probability as he does. [I UPDATED THIS ABOVE]  
  
# p\_PFSD\_SoC <- PFS TO DEAD  
# p\_FP\_SoC <- PFS TO OS  
#   
#   
#   
# p\_PFSD\_Exp <- PFS TO DEAD  
# p\_FP\_Exp <- PFS TO OS  
#   
#   
# p\_FD\_SoC <- Probability of dying when in OS.  
# p\_FD\_Exp <- Probability of dying when in OS.  
  
  
# I could probably apply parametric survival models to OS data, as all individuals from the OS curve were under standard of care, and get free to dead that way from the data.  
  
  
  
# PFS -> PFS:  
  
#p\_PFS\_SoC <- (1 - p\_FA1\_STD)\* (1 - p\_FA2\_STD)\* (1 - p\_FA3\_STD)\* (1 - p\_FP\_SoC) \* (1 - p\_PFSD\_SoC)   
#p\_PFS\_Exp <- (1 - p\_PFSD\_Exp) \* (1 - p\_FP\_Exp)\* (1 - p\_FA1\_EXPR)\* (1 - p\_FA2\_EXPR)\* (1 - p\_FA3\_EXPR)  
  
# First, I need to create progression free survival probabilities under standard of care.  
  
# It's 1- p\_PFSD\_SoC, because if you're not going from healthy to dead then you're staying in healthy (i.e. you're not going from PFS to Dead, so you're staying in PFS) so this captures all the people leftover in PFS after those going to dead, and again for 1- p\_FP\_SoC, if you're not going from healthy to sick, the only other way out of healthy per the transition probability matrix, then you're staying in PFS, so 1- p\_FP\_SoC takes away all the people who went from healthy to sick, and leaves behind all the people who stayed in healthy (or takes away all the people who went to OS and leaves behind all the people that went to PFS per my model). I also add in the three adverse events as well.  
  
  
# PFS -> OS:  
#   
# p\_PFS\_OS\_SoC <- (1 - p\_PFSD\_SoC) \* (1 - p\_FA1\_STD)\* (1 - p\_FA2\_STD)\* (1 - p\_FA3\_STD)\* p\_FP\_SoC  
#   
#   
# p\_PFS\_OS\_Exp <- (1 - p\_PFSD\_Exp) \* (1 - p\_FA1\_EXPR)\* (1 - p\_FA2\_EXPR)\* (1 - p\_FA3\_EXPR) \* p\_FP\_Exp  
#   
#   
  
  
  
  
# We have to get the probability for going from PFS to OS, by getting all the people left in PFS after the people who went to death were gone, and then we multiply this probability by the probability of going from healthy to sick, or PFS to OS, because it's the probability conditional on being alive, the probability conditional on being left in the PFS state after the other people who went to the death state are gone, so the probability of going from PFS to OS, conditional on surviving.  
  
  
# p\_PFS\_OS\_SoC is the probability of transitioning from healthy to sick, conditional on surviving, so it's defined as a conditional probability. So, what the Markov model wants is : m\_P\_SoC["PFS", "OS"] what's the overall probability of transitioning from PFS to OS, and so that is actually not a conditional probability, which is why we multiply the probability of surviving (1 - p\_HD) \* p\_HS\_SoC by the probability of going to OS conditional on surviving, because the transition in the model should be the marginal not the conditional, i.e. you want to have the end probability.  
  
  
#So, because you took a probability that was conditional from the literature, or digitised a PFS curve to get a transition probability - and those probabilities come from people who were necessarily still in the PFS state when you digitise a PFS curve, so those are transition probabilities conditional on being in the PFS (or healthy) state - you need to do the multiplication to give you a probability you can put in your transition matrix that reflects that the probability you are working with is conditional.  
  
  
# # PFS -> DEAD:  
#   
# p\_PFSD\_SoC <- (p\_PFS\_SoC) \* (p\_PFSD\_SoC)  
# p\_PFSD\_Exp <- (p\_PFS\_Exp) \* (p\_PFSD\_Exp)  
  
  
  
# Adverse event transition probabilities have all been conditional, this is to reflect that probabilities on transitioning from adverse event to death will necessarily be conditional probabilities - as they come from individuals who were studied in the adverse event group - so they are necessarily probabilities conditional on experiencing the adverse event. Likewise, the probability of experiencing the adverse event when under treatment is conditional on being in PFS for the treatment in the first place, because it will come from reports of adverse events in studies of the treatments given that we look at.   
  
  
# Which means conditional probabilities kind of don't matter too much, because the probability won't be applied to anyone who isnt in the state it doesnt necessarily need to be conditional after all. And I guess it would have never been applied to anyone who wasnt in the state, because probabilities are only applied to people who are in the state when the transition probability matrix is multiplied by the cohort trace of people in the state.  
  
  
  
# p\_FA1\_SoC <- (p\_PFS\_SoC) \* p\_FA1\_STD  
  
  
# Probability of AE1 when PFS, conditional on surviving, under standard of care  
  
  
  
#p\_A1D\_SoC <- 0.001   
  
# Probability of going from AE1 to death, at the moment this is just a dummy value, I expect this to be a really low value source from the literature.  
  
  
  
#p\_A1F\_SoC <- 1-p\_A1D\_SoC  
  
# Probability of returning from AE1 to PFS is 100% minus the people who have gone into the dead state.  
  
  
  
  
# I repeat this for the other adverse events and for both standard of care and the experimental (novel) treatment:  
  
  
# PFS -> PFS:  
#   
# p\_PFS\_SoC <- (1 - p\_FA1\_STD)\* (1 - p\_FA2\_STD)\* (1 - p\_FA3\_STD)\* (1 - p\_PFSOS\_SoC) \* (1 - p\_PFSD\_SoC)   
# p\_PFS\_Exp <- (1 - p\_PFSD\_Exp) \* (1 - p\_PFSOS\_Exp)\* (1 - p\_FA1\_EXPR)\* (1 - p\_FA2\_EXPR)\* (1 - p\_FA3\_EXPR)  
#   
#   
# # Probability of A2 when PFS, conditional on surviving, under standard of care  
# p\_FA1\_SoC <- (p\_PFS\_SoC) \* p\_FA1\_STD  
# # Probability of going from A2 to death  
# p\_A1D\_SoC <- 0.001   
#   
#   
# # Probability of A2 when PFS, conditional on surviving, under standard of care  
# p\_FA2\_SoC <- (p\_PFS\_SoC) \* p\_FA2\_STD  
# # Probability of going from A2 to death, at the moment this is just a dummy value, I expect this to be a really low value source from the literature.  
# p\_A2D\_SoC <- 0.001   
# # Probability of returning from A2 to PFS is 100% minus the people who have gone into the dead state.  
# p\_A2F\_SoC <- 1-p\_A2D\_SoC  
#   
# # Probability of A3 when PFS, conditional on surviving, under standard of care  
# p\_FA3\_SoC <- (p\_PFS\_SoC) \* p\_FA3\_STD  
# # Probability of going from A3 to death, at the moment this is just a dummy value, I expect this to be a really low value source from the literature.  
# p\_A3D\_SoC <- 0.001   
# # Probability of returning from A3 to PFS is 100% minus the people who have gone into the dead state.  
# p\_A3F\_SoC <- 1-p\_A3D\_SoC  
#   
#   
#   
#   
#   
#   
# # Probability of A1 when PFS, conditional on surviving, under standard of care  
# p\_FA1\_EXPR <- (p\_PFS\_Exp) \* p\_FA1\_EXPR  
# # Probability of going from A1 to death, at the moment this is just a dummy value, I expect this to be a really low value source from the literature.  
# p\_A1D\_Exp <- 0.001   
# # Probability of returning from A1 to PFS is 100% minus the people who have gone into the dead state.  
# p\_A1F\_Exp <- 1-p\_A1D\_Exp  
#   
# # Probability of A2 when PFS, conditional on surviving, under standard of care  
# p\_FA2\_Exp <- (p\_PFS\_Exp) \* p\_FA2\_EXPR  
# # Probability of going from A2 to death, at the moment this is just a dummy value, I expect this to be a really low value source from the literature.  
# p\_A2D\_Exp <- 0.001   
# # Probability of returning from A2 to PFS is 100% minus the people who have gone into the dead state.  
# p\_A2F\_Exp <- 1-p\_A2D\_Exp  
#   
# # Probability of A3 when PFS, conditional on surviving, under standard of care  
# p\_FA3\_Exp <- (p\_PFS\_Exp) \* p\_FA3\_EXPR  
# # Probability of going from A3 to death, at the moment this is just a dummy value, I expect this to be a really low value source from the literature.  
# p\_A3D\_Exp <- 0.001   
# # Probability of returning from A3 to PFS is 100% minus the people who have gone into the dead state.  
# p\_A3F\_Exp <- 1-p\_A3D\_Exp  
#   
#   
# # These all have to be conditional on survival, i.e. conditional on being in PFS, this is to ensure that I am only applying adverse events to the people still in the PFS, after transitions etc., have happened. --> This is not how conditional probabilities work, I have the correct interpretation of them elsewhere in this code file.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
## Health State Values (AKA State rewards)  
# Costs and utilities   
# Basically the outcomes we are interested in coming out of this model, so we'll look at the cohorts costs over the time horizon and the quality adjusted life years in our cohort over this time horizon.  
  
# Costs  
  
c\_PFS\_Folfox <-307.81   
c\_PFS\_Bevacizumab <-2580.38   
c\_OS\_Folfiri <-326.02   
administration\_cost <- 365.00   
  
  
c\_F\_SoC <- administration\_cost + c\_PFS\_Folfox # cost of one cycle in PFS state under standard of care  
c\_F\_Exp <- administration\_cost + c\_PFS\_Folfox + c\_PFS\_Bevacizumab # cost of one cycle in PFS state under the experimental treatment   
#c\_F\_Ex\_trtA <- 1000 # cost of one cycle in PFS state under the experimental treatment A   
#c\_F\_Ex\_trtB <- 1000 # cost of one cycle in PFS state under the experimental treatment  
c\_P <- c\_OS\_Folfiri + administration\_cost# cost of one cycle in progression state (I assume in OS everyone gets the same treatment so it costs everyone the same to be treated).  
c\_D <- 0 # cost of one cycle in dead state  
  
  
  
# Above is the cost for each state, PFS, OS and dead,  
  
# c\_trtA <- 800 # cost of EPI Assay in PFS state (ONCE OFF COST)  
# c\_trtB <- 1500 # cost of HDX Assay in PFS state (ONCE OFF COST)  
#   
# # We make the cost of the assays above so that when we have treatment strategies we can add this cost of treatment to anyone whose being treated when they receive the treatment.  
  
  
# Then we define the utilities per health states.  
  
  
u\_F <- 0.850 # utility when PFS   
u\_P <- 0.650 # utility when OS  
u\_D <- 0 # utility when dead  
  
  
  
  
# Discounting factors  
d\_c <- 0.04   
# discount rate for costs (per year)  
d\_e <- 0.04   
# discount rate for QALYs (per year)  
  
# discount rate per cycle equal discount of costs and QALYs by 4%  
  
# Discount weight (equal discounting is assumed for costs and effects)  
  
# Actually I've updated this and do this later on now  
  
# v\_dwc <- 1 / (1 + d\_c) ^ (0:n\_cycle)   
# v\_dwe <- 1 / (1 + d\_e) ^ (0:n\_cycle)   
  
# So, we create a discount weight vector above, to understand the way this works I'll have to return to my York notes on discounting  
  
p\_PFSD\_SoC

## [1] 0.004708444 0.007309584 0.008772251 0.009880615 0.010795647 0.011585361  
## [7] 0.012285903 0.012919101 0.013499287 0.014036442 0.014537832 0.015008933  
## [13] 0.015453987 0.015876356 0.016278753 0.016663408 0.017032173 0.017386609  
## [19] 0.017728046 0.018057624 0.018376331 0.018685030 0.018984478 0.019275345  
## [25] 0.019558225 0.019833651 0.020102098 0.020363996 0.020619734 0.020869666  
## [31] 0.021114111 0.021353365 0.021587697 0.021817354 0.022042564 0.022263539  
## [37] 0.022480473 0.022693548 0.022902933 0.023108783 0.023311247 0.023510460  
## [43] 0.023706551 0.023899640 0.024089841 0.024277261 0.024461998 0.024644148  
## [49] 0.024823800 0.025001039 0.025175944 0.025348592 0.025519054 0.025687399  
## [55] 0.025853692 0.026017994 0.026180364 0.026340859 0.026499532 0.026656434  
## [61] 0.026811615 0.026965120 0.027116995 0.027267282 0.027416023 0.027563257  
## [67] 0.027709021 0.027853353 0.027996286 0.028137856 0.028278093 0.028417029  
## [73] 0.028554693 0.028691116 0.028826324 0.028960344 0.029093202 0.029224923  
## [79] 0.029355532 0.029485051 0.029613503 0.029740910 0.029867293 0.029992673  
## [85] 0.030117069 0.030240500 0.030362985 0.030484542 0.030605189 0.030724942  
## [91] 0.030843817 0.030961832 0.031079000 0.031195338 0.031310860 0.031425580  
## [97] 0.031539512 0.031652669 0.031765065 0.031876712 0.031987623 0.032097809  
## [103] 0.032207283 0.032316056 0.032424139 0.032531543 0.032638278 0.032744355  
## [109] 0.032849784 0.032954575 0.033058737 0.033162280 0.033265213 0.033367545  
## [115] 0.033469284 0.033570440 0.033671020 0.033771033 0.033870486 0.033969387  
## [121] 0.034067745 0.034165565 0.034262857 0.034359626 0.034455880 0.034551625  
## [127] 0.034646869 0.034741618 0.034835878 0.034929655 0.035022956 0.035115786  
## [133] 0.035208152 0.035300059 0.035391514 0.035482521 0.035573085 0.035663213  
## [139] 0.035752910 0.035842180 0.035931029 0.036019462 0.036107483

p\_PFSD\_Exp

## [1] 0.003063016 0.004757328 0.005710751 0.006433555 0.007030492 0.007545832  
## [7] 0.008003102 0.008416512 0.008795392 0.009146241 0.009473790 0.009781605  
## [13] 0.010072449 0.010348510 0.010611556 0.010863039 0.011104166 0.011335954  
## [19] 0.011559268 0.011774851 0.011983348 0.012185320 0.012381261 0.012571607  
## [25] 0.012756746 0.012937023 0.013112750 0.013284206 0.013451645 0.013615297  
## [31] 0.013775371 0.013932058 0.014085535 0.014235963 0.014383490 0.014528255  
## [37] 0.014670383 0.014809994 0.014947197 0.015082094 0.015214781 0.015345348  
## [43] 0.015473877 0.015600448 0.015725134 0.015848005 0.015969126 0.016088558  
## [49] 0.016206360 0.016322587 0.016437291 0.016550521 0.016662325 0.016772748  
## [55] 0.016881830 0.016989613 0.017096135 0.017201433 0.017305542 0.017408494  
## [61] 0.017510323 0.017611057 0.017710728 0.017809362 0.017906986 0.018003626  
## [67] 0.018099307 0.018194052 0.018287885 0.018380826 0.018472898 0.018564119  
## [73] 0.018654511 0.018744092 0.018832879 0.018920890 0.019008143 0.019094654  
## [79] 0.019180437 0.019265509 0.019349884 0.019433577 0.019516600 0.019598968  
## [85] 0.019680694 0.019761789 0.019842267 0.019922138 0.020001414 0.020080107  
## [91] 0.020158226 0.020235783 0.020312787 0.020389249 0.020465177 0.020540581  
## [97] 0.020615471 0.020689854 0.020763740 0.020837137 0.020910052 0.020982495  
## [103] 0.021054471 0.021125990 0.021197057 0.021267681 0.021337868 0.021407625  
## [109] 0.021476958 0.021545875 0.021614380 0.021682481 0.021750183 0.021817493  
## [115] 0.021884415 0.021950955 0.022017119 0.022082913 0.022148340 0.022213407  
## [121] 0.022278119 0.022342480 0.022406494 0.022470168 0.022533504 0.022596508  
## [127] 0.022659184 0.022721537 0.022783570 0.022845287 0.022906693 0.022967792  
## [133] 0.023028586 0.023089081 0.023149280 0.023209186 0.023268803 0.023328135  
## [139] 0.023387185 0.023445955 0.023504451 0.023562674 0.023620628

p\_PFSOS\_SoC

## [1] 0.01484724 0.02399843 0.02932054 0.03340742 0.03681073 0.03976670  
## [7] 0.04240199 0.04479366 0.04699259 0.04903438 0.05094508 0.05274438  
## [13] 0.05444758 0.05606686 0.05761207 0.05909131 0.06051136 0.06187791  
## [19] 0.06319584 0.06446934 0.06570203 0.06689710 0.06805734 0.06918523  
## [25] 0.07028297 0.07135252 0.07239566 0.07341399 0.07440894 0.07538184  
## [31] 0.07633389 0.07726618 0.07817972 0.07907543 0.07995417 0.08081674  
## [37] 0.08166385 0.08249621 0.08331443 0.08411910 0.08491079 0.08569001  
## [43] 0.08645723 0.08721292 0.08795750 0.08869138 0.08941493 0.09012851  
## [49] 0.09083246 0.09152710 0.09221274 0.09288966 0.09355813 0.09421841  
## [55] 0.09487076 0.09551541 0.09615257 0.09678247 0.09740531 0.09802128  
## [61] 0.09863057 0.09923336 0.09982982 0.10042011 0.10100440 0.10158282  
## [67] 0.10215553 0.10272267 0.10328436 0.10384074 0.10439193 0.10493805  
## [73] 0.10547921 0.10601553 0.10654711 0.10707406 0.10759647 0.10811444  
## [79] 0.10862806 0.10913743 0.10964263 0.11014374 0.11064084 0.11113402  
## [85] 0.11162334 0.11210889 0.11259074 0.11306895 0.11354359 0.11401473  
## [91] 0.11448243 0.11494675 0.11540776 0.11586550 0.11632004 0.11677144  
## [97] 0.11721974 0.11766499 0.11810726 0.11854658 0.11898301 0.11941659  
## [103] 0.11984737 0.12027539 0.12070069 0.12112332 0.12154333 0.12196074  
## [109] 0.12237559 0.12278794 0.12319781 0.12360524 0.12401026 0.12441291  
## [115] 0.12481323 0.12521125 0.12560699 0.12600050 0.12639180 0.12678092  
## [121] 0.12716789 0.12755275 0.12793551 0.12831621 0.12869488 0.12907153  
## [127] 0.12944621 0.12981893 0.13018971 0.13055859 0.13092558 0.13129071  
## [133] 0.13165400 0.13201548 0.13237517 0.13273309 0.13308925 0.13344369  
## [139] 0.13379642 0.13414746 0.13449683 0.13484455 0.13519064

p\_PFSOS\_Exp

## [1] 0.01012026 0.01638227 0.02003273 0.02284029 0.02518117 0.02721649  
## [7] 0.02903271 0.03068241 0.03220033 0.03361079 0.03493156 0.03617610  
## [13] 0.03735488 0.03847620 0.03954680 0.04057223 0.04155710 0.04250533  
## [19] 0.04342023 0.04430468 0.04516116 0.04599185 0.04679865 0.04758326  
## [25] 0.04834719 0.04909178 0.04981825 0.05052769 0.05122109 0.05189935  
## [31] 0.05256329 0.05321367 0.05385117 0.05447643 0.05509003 0.05569251  
## [37] 0.05628438 0.05686611 0.05743812 0.05800083 0.05855461 0.05909981  
## [43] 0.05963677 0.06016579 0.06068718 0.06120120 0.06170812 0.06220818  
## [49] 0.06270162 0.06318865 0.06366949 0.06414432 0.06461334 0.06507673  
## [55] 0.06553465 0.06598726 0.06643473 0.06687719 0.06731480 0.06774767  
## [61] 0.06817594 0.06859973 0.06901916 0.06943434 0.06984538 0.07025239  
## [67] 0.07065545 0.07105467 0.07145014 0.07184195 0.07223018 0.07261492  
## [73] 0.07299624 0.07337421 0.07374892 0.07412043 0.07448882 0.07485414  
## [79] 0.07521646 0.07557585 0.07593235 0.07628604 0.07663697 0.07698518  
## [85] 0.07733073 0.07767368 0.07801408 0.07835196 0.07868738 0.07902038  
## [91] 0.07935100 0.07967930 0.08000530 0.08032905 0.08065059 0.08096995  
## [97] 0.08128718 0.08160230 0.08191536 0.08222638 0.08253541 0.08284246  
## [103] 0.08314758 0.08345079 0.08375213 0.08405162 0.08434929 0.08464517  
## [109] 0.08493929 0.08523166 0.08552233 0.08581130 0.08609862 0.08638429  
## [115] 0.08666835 0.08695082 0.08723172 0.08751107 0.08778889 0.08806520  
## [121] 0.08834003 0.08861339 0.08888531 0.08915580 0.08942488 0.08969256  
## [127] 0.08995888 0.09022384 0.09048747 0.09074977 0.09101077 0.09127048  
## [133] 0.09152892 0.09178610 0.09204204 0.09229675 0.09255025 0.09280256  
## [139] 0.09305368 0.09330363 0.09355242 0.09380008 0.09404660

Discount rate for costs and utilities I need to return to it more generally with the notes I wrote from York on discounting. Also I set up 3 strategies, i.e. standard of care, EPI Assay and HDX Assay, to reflect the two assays under study, but it would be very easy to change this to just 2 strategies above.

I also want to add in the costing per the York model, as this broke costs down before adding them, although I can compare this to the York approach to costing in their published article, i.e. are costs in that article combined before they are added to the model or afterwards?

There are some things that I would like to appear in the code chunks, but not in the knitted document. To do this I can go to:

<https://stackoverflow.com/questions/47710427/how-to-show-code-but-hide-output-in-rmarkdown>

and

<https://stackoverflow.com/questions/48286722/rmarkdown-how-to-show-partial-output-from-chunk?rq=1>

Although I can probably just use:

knitr::opts\_chunk$set(echo = TRUE, warning = FALSE, message = FALSE, eval = T)

But set echo = false

Or even better, click the gear on each code chunk and decide if I would like that code chunk to show things or not.

If I was interested in how to add adverse events to the model, Eva describes how to create an additional state that is Sick+AdverseEvent here:

C:- Royal College of Surgeons in IrelandResources-Effectiveness and Decision Modeling using R Workshop \_ DARTH\_25Session RecordingSession Recording August 25th WITH CHAT.mkv

## Draw the state-transition cohort model

#   
# diag\_names\_states <- c("PFS", "OS", "Dead")   
#   
# m\_P\_diag <- matrix(0, nrow = n\_states, ncol = n\_states, dimnames = list(diag\_names\_states, diag\_names\_states))  
#   
# m\_P\_diag["PFS", "PFS" ] = ""  
# m\_P\_diag["PFS", "OS" ] = ""   
# m\_P\_diag["PFS", "Dead" ] = ""  
# m\_P\_diag["OS", "OS" ] = ""  
# m\_P\_diag["OS", "Dead" ] = ""  
# m\_P\_diag["Dead", "Dead" ] = ""  
# layout.fig <- c(1, 3, 2) # <- changing the numbers here changes the diagram layout, so mess with these until I'm happy. It basically decides how many bubbles will be on each level, so here 1 bubble, followed by 3 bubbles, followed by 2 bubbles, per the diagram for 1, 3, 2.  
# plotmat(t(m\_P\_diag), t(layout.fig), self.cex = 0.5, curve = 0, arr.pos = 0.64,   
# latex = T, arr.type = "curved", relsize = 0.85, box.prop = 0.9,   
# cex = 0.8, box.cex = 0.7, lwd = 0.6, main = "Figure 1")

# 04 Define and initialize matrices and vectors

After setting up our parameters above, we initialise our structure below.

This is where we will store all of the model output, and all the things that we need to track over time as we are simulating the progression of this cohort through this disease process.

## 04.1 Cohort trace

# WHEN COMING BACK TO COMPARE: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\Parametric Survival Analysis\ISPOR WEBINAR Health Economic Modelling in R\ISPOR\_webinar\_R-master\ISPOR\_webinar\_R-master\oncologySemiMarkov\_illustration to this Rmarkdown document, a big difference is that this document creates a cycle 0 - whereas the comparison from the ISPOR\_webinar\_R uses -1 to get into cycle 0 where necessary. I have decided to follow the ISPOR way, because I am interested in duplicating their parametric analysis, so I need to bear this difference in mind as I go through this document.  
  
# Also, continue from Verbose in this and in Koens file  
  
# Markov cohort trace matrix ----  
  
# Initialize matrices to store the Markov cohort traces for each strategy  
  
# - note that the number of rows is n\_cycle + 1, because R doesn't use index 0 (i.e. cycle 0) --> What we mean here, is that when we do our calculations later they need to be for cycle-1 to reflect cycle 0.  
m\_M\_SoC <- m\_M\_Exp <- matrix(  
 data = NA,   
 nrow = n\_cycle,   
 ncol = n\_states,   
 dimnames = list(paste('Cycle', 1:n\_cycle), v\_names\_states)  
)  
  
## Initial state vector  
# We create an inital vector where people start, with everyone (1 = 100% of people) starting in PFS below:  
# v\_s\_init <- c("PFS" = 1, "OS" = 0, "Dead" = 0)  
# v\_s\_init  
  
# There are cases where you can have an initial illness prevalence, so you would start some people in the sick state and some people in the healthy state, but above we're looking at people with mCRC, so we'll start everyone in PFS.  
  
  
## Initialize cohort trace for cSTM (cohort state transition model) for all strategies (the strategies are the treatment strategies SOC, treatment A and Treatment B).  
# So, basically we are creating a matrix to trace how the cohort is distributed across the health states, over time.   
  
# A matrix is necessary because there are basically two dimensions to this, the number of time cycles, which will be our rows, and then the number of states - to know which proportion of our cohort is in each state at each time:  
  
# m\_M\_SoC <- matrix(0,   
# nrow = (n\_cycles + 1), ncol = n\_states,   
# dimnames = list(v\_names\_cycles, v\_names\_states))  
# Above instead of having to bother with -1 throughout the analysis they create a cycle 0.  
  
# Store the initial state vector in the first row of the cohort trace  
# m\_M\_SoC[1, ] <- v\_s\_init  
## Initialize cohort traces  
## So, above I made the cohort trace for standard of care, because in my analysis all my patients start in the PFS state, I can duplicate that below to create the cohort trace for treatment A and treatment B.  
# m\_M\_trtA <- m\_M\_trtB <- m\_M\_SoC # structure and initial states remain the same  
  
# This gives us three matrices, m\_M\_trtA, m\_M\_trtB and m\_M\_SoC, that we can fill in with our simulations of how patients transitions between health states under each treatment strategy.  
  
# In the first row of the markov matrix [1, ] put the value at the far end, i.e. "<-1" and "<-0" under the colum "PFS" [ , "PFS"], repeating this for "OS", "AE1", "AE2" "AE3" and "Dead".  
  
  
# Specifying the initial state for the cohorts (all patients start in PFS)  
m\_M\_SoC[1, "PFS"] <- m\_M\_Exp[1, "PFS"] <- 1  
m\_M\_SoC[1, "OS"] <- m\_M\_Exp[1, "OS"] <- 0  
m\_M\_SoC[1, "Dead"]<- m\_M\_Exp[1, "Dead"] <- 0  
  
# Inspect whether properly defined  
head(m\_M\_SoC)

## PFS OS Dead  
## Cycle 1 1 0 0  
## Cycle 2 NA NA NA  
## Cycle 3 NA NA NA  
## Cycle 4 NA NA NA  
## Cycle 5 NA NA NA  
## Cycle 6 NA NA NA

head(m\_M\_Exp)

## PFS OS Dead  
## Cycle 1 1 0 0  
## Cycle 2 NA NA NA  
## Cycle 3 NA NA NA  
## Cycle 4 NA NA NA  
## Cycle 5 NA NA NA  
## Cycle 6 NA NA NA

#head(m\_M\_Exp\_trtB)

## 04.2 Transition probability matrix

## If there were time varying transition probabilities, i.e. the longer you are in the model there are changes in your transition probability into death as you get older, etc., you would build a transition probability array, rather than a transition probability matrix, per:   
  
# 04.2 of:  
  
# "C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modeling for Public Health\_DARTH\5\_Nov\_29\4\_Cohort state-transition models (cSTM) - time-dependent models\_material\Markov\_3state\_time"  
  
# with the 1hour: 02minute mark of: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_24\Live Session Recording\Live Session Recording August 24th.mp4  
  
  
## Initialize transition probability matrix, [i.e. build the framework or empty scaffolding of the transition probability matrix]  
# all transitions to a non-death state are assumed to be conditional on survival  
# - starting with standard of care  
# - note that these are now 3-dimensional matrices because we are including time.  
   
  
# m\_P\_SoC <- matrix(0,  
# nrow = n\_states, ncol = n\_states,  
# dimnames = list(v\_names\_states, v\_names\_states)) # define row and column names  
# m\_P\_SoC  
  
  
# Initialize matrices for the transition probabilities  
# - note that these are now 3-dimensional matrices (so, above we originally included dim = nrow and ncol, but now we also include n\_cycle - i.e. the number of cycles).  
# - starting with standard of care  
m\_P\_SoC <- array(  
 data = 0,  
 dim = c(n\_states, n\_states, n\_cycle),  
 dimnames = list(v\_names\_states, v\_names\_states, paste0("Cycle", 1:n\_cycle))  
 # define row and column names - then name each array after which cycle it's for, i.e. cycle 1 all the way through to cycle 143. So Cycle 1 will have all of our patients in PFS, while cycle 143 will have most people in the dead state.  
)  
  
head(m\_P\_SoC)

## , , Cycle1  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle2  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle3  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle4  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle5  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle6  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle7  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle8  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle9  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle10  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle11  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle12  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle13  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle14  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle15  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle16  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle17  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle18  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle19  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle20  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle21  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle22  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle23  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle24  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle25  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle26  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle27  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle28  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle29  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle30  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle31  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle32  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle33  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle34  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle35  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle36  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle37  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle38  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle39  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle40  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle41  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle42  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle43  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle44  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle45  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle46  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle47  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle48  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle49  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle50  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle51  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle52  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle53  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle54  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle55  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle56  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle57  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle58  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle59  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle60  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle61  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle62  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle63  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle64  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle65  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle66  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle67  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle68  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle69  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle70  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle71  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle72  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle73  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle74  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle75  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle76  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle77  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle78  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle79  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle80  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle81  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle82  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle83  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle84  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle85  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle86  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle87  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle88  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle89  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle90  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle91  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle92  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle93  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle94  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle95  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle96  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle97  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle98  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle99  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle100  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle101  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle102  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle103  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle104  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle105  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle106  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle107  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle108  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle109  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle110  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle111  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle112  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle113  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle114  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle115  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle116  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle117  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle118  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle119  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle120  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle121  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle122  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle123  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle124  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle125  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle126  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle127  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle128  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle129  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle130  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle131  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle132  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle133  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle134  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle135  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle136  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle137  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle138  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle139  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle140  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle141  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle142  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle143  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0

m\_P\_Exp <- array(  
 data = 0,  
 dim = c(n\_states, n\_states, n\_cycle),  
 dimnames = list(v\_names\_states, v\_names\_states, paste0("Cycle", 1:n\_cycle))  
 # define row and column names - then name each array after which cycle it's for, i.e. cycle 1 all the way through to cycle 143. So Cycle 1 will have all of our patients in PFS, while cycle 143 will have most people in the dead state.  
)  
  
head(m\_P\_Exp)

## , , Cycle1  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle2  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle3  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle4  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle5  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle6  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle7  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle8  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle9  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle10  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle11  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle12  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle13  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle14  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle15  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle16  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle17  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle18  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle19  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle20  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle21  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle22  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle23  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle24  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle25  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle26  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle27  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle28  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle29  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle30  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle31  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle32  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle33  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle34  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle35  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle36  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle37  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle38  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle39  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle40  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle41  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle42  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle43  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle44  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle45  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle46  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle47  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle48  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle49  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle50  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle51  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle52  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle53  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle54  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle55  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle56  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle57  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle58  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle59  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle60  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle61  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle62  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle63  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle64  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle65  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle66  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle67  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle68  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle69  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle70  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle71  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle72  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle73  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle74  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle75  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle76  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle77  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle78  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle79  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle80  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle81  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle82  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle83  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle84  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle85  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle86  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle87  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle88  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle89  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle90  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle91  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle92  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle93  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle94  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle95  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle96  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle97  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle98  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle99  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle100  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle101  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle102  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle103  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle104  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle105  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle106  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle107  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle108  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle109  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle110  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle111  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle112  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle113  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle114  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle115  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle116  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle117  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle118  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle119  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle120  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle121  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle122  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle123  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle124  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle125  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle126  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle127  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle128  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle129  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle130  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle131  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle132  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle133  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle134  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle135  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle136  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle137  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle138  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle139  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle140  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle141  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle142  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0  
##   
## , , Cycle143  
##   
## PFS OS Dead  
## PFS 0 0 0  
## OS 0 0 0  
## Dead 0 0 0

## 04.3 Fill in the transition probability matrix:

# It looks like I will need to decide if I have to go back to the start of putting everything in the model and put stuff in for not just the experimental strategy, but for experimental strategy 1 AND experimental strategy 2 individually.  
  
#   
# AE1MARK <- p\_FA1\_STD\* (1 - p\_FA2\_STD)\* (1 - p\_FA3\_STD)\* (1 - p\_FP\_SoC) \* (1 - p\_PFSD\_SoC)  
# AE2MARK <- (1 - p\_FA1\_STD)\* p\_FA2\_STD\* (1 - p\_FA3\_STD)\* (1 - p\_FP\_SoC) \* (1 - p\_PFSD\_SoC)  
# AE3MARK <- (1 - p\_FA1\_STD)\* (1 - p\_FA2\_STD)\* p\_FA3\_STD\* (1 - p\_FP\_SoC) \* (1 - p\_PFSD\_SoC)  
  
# If I really struggle with the transition probabilities below, I could build them from here alternatively: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Applied Cost Effectiveness Modeling with R\rcea-rscripts\01-markov-cohort  
  
# A better way to fill this may be here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Model Calibration in R\UNZIP to Working Directory\_Calibration Participant Materials\ISPOR Calibration Participant Materials\SickSicker\_MarkovModel\_Function.R  
  
# This also gives advice on filling this in: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\manuscript\Appendix\_code\_walkthrough\_Markov\_Tutorial\_Part2  
  
# Andy gave me advice on how to add adverse event disutility and costs, but I could also think about the strategy taken here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\R\Functions\_cSTM\_time\_dep\_simulation  
  
# Setting the transition probabilities from PFS based on the model parameters  
 # So, when individuals are in PFS what are their probabilities of going into the other states that they can enter from PFS?  
  
m\_P\_SoC["PFS", "PFS",]<- (1 -p\_PFSOS\_SoC) \* (1 - p\_PFSD\_SoC)  
m\_P\_SoC["PFS", "OS",]<- p\_PFSOS\_SoC\*(1 - p\_PFSD\_SoC)  
m\_P\_SoC["PFS", "Dead",]<-p\_PFSD\_SoC  
  
# m\_P\_SoC["PFS", "PFS", ] <- 1 - (p\_PFSD\_SoC + p\_FP\_SoC)  
# m\_P\_SoC["PFS", "OS", ] <- 1 - (1-p\_FP\_SoC + p\_PFSD\_SoC)  
# m\_P\_SoC["PFS", "Dead", ] <- 1 - (p\_FP\_SoC + 1 - p\_PFSD\_SoC)  
  
  
#m\_P\_SoC["PFS", "PFS", ] <- 1 - (p\_PFSD\_SoC + p\_FP\_SoC)  
#m\_P\_SoC["PFS", "OS", ] <- p\_FP\_SoC\* (1 - p\_PFSD\_SoC)  
#m\_P\_SoC["PFS", "Dead", ] <- (1 - p\_FP\_SoC) \* (p\_PFSD\_SoC)  
  
# Setting the transition probabilities from OS  
m\_P\_SoC["OS", "OS", ] <- 1 - P\_OSD\_SoC  
m\_P\_SoC["OS", "Dead", ] <- P\_OSD\_SoC  
  
  
# Setting the transition probabilities from Dead  
m\_P\_SoC["Dead", "Dead", ] <- 1  
  
#   
# # Setting the transition probabilities from AE1  
# m\_P\_SoC["AE1", "PFS", ] <- p\_A1F\_SoC  
# m\_P\_SoC["AE1", "Dead", ] <- p\_A1D\_SoC  
#   
# # Setting the transition probabilities from AE2  
# m\_P\_SoC["AE2", "PFS", ] <- p\_A2F\_SoC  
# m\_P\_SoC["AE2", "Dead", ] <- p\_A2D\_SoC  
#   
# # Setting the transition probabilities from AE3  
# m\_P\_SoC["AE3", "PFS", ] <- p\_A3F\_SoC  
# m\_P\_SoC["AE3", "Dead", ] <- p\_A3D\_SoC  
  
  
m\_P\_SoC

## , , Cycle1  
##   
## PFS OS Dead  
## PFS 0.9805142 0.01477733 0.004708444  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle2  
##   
## PFS OS Dead  
## PFS 0.9688674 0.02382301 0.007309584  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle3  
##   
## PFS OS Dead  
## PFS 0.9621644 0.02906334 0.008772251  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle4  
##   
## PFS OS Dead  
## PFS 0.9570421 0.03307733 0.009880615  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle5  
##   
## PFS OS Dead  
## PFS 0.952791 0.03641334 0.01079565  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle6  
##   
## PFS OS Dead  
## PFS 0.9491087 0.03930599 0.01158536  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle7  
##   
## PFS OS Dead  
## PFS 0.9458331 0.04188104 0.0122859  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle8  
##   
## PFS OS Dead  
## PFS 0.9428659 0.04421497 0.0129191  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle9  
##   
## PFS OS Dead  
## PFS 0.9401425 0.04635822 0.01349929  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle10  
##   
## PFS OS Dead  
## PFS 0.9376174 0.04834612 0.01403644  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle11  
##   
## PFS OS Dead  
## PFS 0.9352577 0.05020445 0.01453783  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle12  
##   
## PFS OS Dead  
## PFS 0.9330383 0.05195274 0.01500893  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle13  
##   
## PFS OS Dead  
## PFS 0.9309399 0.05360615 0.01545399  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle14  
##   
## PFS OS Dead  
## PFS 0.9289469 0.05517672 0.01587636  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle15  
##   
## PFS OS Dead  
## PFS 0.927047 0.05667421 0.01627875  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle16  
##   
## PFS OS Dead  
## PFS 0.9252299 0.05810665 0.01666341  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle17  
##   
## PFS OS Dead  
## PFS 0.9234871 0.05948072 0.01703217  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle18  
##   
## PFS OS Dead  
## PFS 0.9218113 0.06080206 0.01738661  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle19  
##   
## PFS OS Dead  
## PFS 0.9201965 0.0620755 0.01772805  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle20  
##   
## PFS OS Dead  
## PFS 0.9186372 0.06330517 0.01805762  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle21  
##   
## PFS OS Dead  
## PFS 0.917129 0.06449467 0.01837633  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle22  
##   
## PFS OS Dead  
## PFS 0.9156678 0.06564712 0.01868503  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle23  
##   
## PFS OS Dead  
## PFS 0.9142502 0.06676531 0.01898448  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle24  
##   
## PFS OS Dead  
## PFS 0.912873 0.06785166 0.01927534  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle25  
##   
## PFS OS Dead  
## PFS 0.9115334 0.06890836 0.01955823  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle26  
##   
## PFS OS Dead  
## PFS 0.910229 0.06993734 0.01983365  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle27  
##   
## PFS OS Dead  
## PFS 0.9089575 0.07094036 0.0201021  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle28  
##   
## PFS OS Dead  
## PFS 0.907717 0.07191898 0.020364  
## OS 0.000000 0.83000000 0.170000  
## Dead 0.000000 0.00000000 1.000000  
##   
## , , Cycle29  
##   
## PFS OS Dead  
## PFS 0.9065056 0.07287465 0.02061973  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle30  
##   
## PFS OS Dead  
## PFS 0.9053217 0.07380865 0.02086967  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle31  
##   
## PFS OS Dead  
## PFS 0.9041637 0.07472217 0.02111411  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle32  
##   
## PFS OS Dead  
## PFS 0.9030303 0.07561629 0.02135337  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle33  
##   
## PFS OS Dead  
## PFS 0.9019203 0.076492 0.0215877  
## OS 0.0000000 0.830000 0.1700000  
## Dead 0.0000000 0.000000 1.0000000  
##   
## , , Cycle34  
##   
## PFS OS Dead  
## PFS 0.9008324 0.07735021 0.02181735  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle35  
##   
## PFS OS Dead  
## PFS 0.8997657 0.07819178 0.02204256  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle36  
##   
## PFS OS Dead  
## PFS 0.898719 0.07901747 0.02226354  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle37  
##   
## PFS OS Dead  
## PFS 0.8976915 0.07982801 0.02248047  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle38  
##   
## PFS OS Dead  
## PFS 0.8966824 0.08062407 0.02269355  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle39  
##   
## PFS OS Dead  
## PFS 0.8956908 0.08140628 0.02290293  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle40  
##   
## PFS OS Dead  
## PFS 0.894716 0.08217521 0.02310878  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle41  
##   
## PFS OS Dead  
## PFS 0.8937573 0.08293142 0.02331125  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle42  
##   
## PFS OS Dead  
## PFS 0.8928141 0.0836754 0.02351046  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle43  
##   
## PFS OS Dead  
## PFS 0.8918858 0.08440763 0.02370655  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle44  
##   
## PFS OS Dead  
## PFS 0.8909718 0.08512856 0.02389964  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle45  
##   
## PFS OS Dead  
## PFS 0.8900715 0.08583862 0.02408984  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle46  
##   
## PFS OS Dead  
## PFS 0.8891845 0.0865382 0.02427726  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle47  
##   
## PFS OS Dead  
## PFS 0.8883103 0.08722766 0.024462  
## OS 0.0000000 0.83000000 0.170000  
## Dead 0.0000000 0.00000000 1.000000  
##   
## , , Cycle48  
##   
## PFS OS Dead  
## PFS 0.8874485 0.08790737 0.02464415  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle49  
##   
## PFS OS Dead  
## PFS 0.8865985 0.08857766 0.0248238  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle50  
##   
## PFS OS Dead  
## PFS 0.8857601 0.08923883 0.02500104  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle51  
##   
## PFS OS Dead  
## PFS 0.8849329 0.0898912 0.02517594  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle52  
##   
## PFS OS Dead  
## PFS 0.8841164 0.09053503 0.02534859  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle53  
##   
## PFS OS Dead  
## PFS 0.8833103 0.09117061 0.02551905  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle54  
##   
## PFS OS Dead  
## PFS 0.8825144 0.09179819 0.0256874  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle55  
##   
## PFS OS Dead  
## PFS 0.8817283 0.092418 0.02585369  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle56  
##   
## PFS OS Dead  
## PFS 0.8809517 0.09303029 0.02601799  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle57  
##   
## PFS OS Dead  
## PFS 0.8801844 0.09363526 0.02618036  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle58  
##   
## PFS OS Dead  
## PFS 0.879426 0.09423314 0.02634086  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle59  
##   
## PFS OS Dead  
## PFS 0.8786764 0.09482411 0.02649953  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle60  
##   
## PFS OS Dead  
## PFS 0.8779352 0.09540838 0.02665643  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle61  
##   
## PFS OS Dead  
## PFS 0.8772023 0.09598613 0.02681161  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle62  
##   
## PFS OS Dead  
## PFS 0.8764774 0.09655752 0.02696512  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle63  
##   
## PFS OS Dead  
## PFS 0.8757603 0.09712274 0.02711699  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle64  
##   
## PFS OS Dead  
## PFS 0.8750508 0.09768193 0.02726728  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle65  
##   
## PFS OS Dead  
## PFS 0.8743487 0.09823526 0.02741602  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle66  
##   
## PFS OS Dead  
## PFS 0.8736539 0.09878287 0.02756326  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle67  
##   
## PFS OS Dead  
## PFS 0.8729661 0.0993249 0.02770902  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle68  
##   
## PFS OS Dead  
## PFS 0.8722852 0.0998615 0.02785335  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle69  
##   
## PFS OS Dead  
## PFS 0.8716109 0.1003928 0.02799629  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle70  
##   
## PFS OS Dead  
## PFS 0.8709433 0.1009189 0.02813786  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle71  
##   
## PFS OS Dead  
## PFS 0.870282 0.1014399 0.02827809  
## OS 0.000000 0.8300000 0.17000000  
## Dead 0.000000 0.0000000 1.00000000  
##   
## , , Cycle72  
##   
## PFS OS Dead  
## PFS 0.869627 0.101956 0.02841703  
## OS 0.000000 0.830000 0.17000000  
## Dead 0.000000 0.000000 1.00000000  
##   
## , , Cycle73  
##   
## PFS OS Dead  
## PFS 0.868978 0.1024673 0.02855469  
## OS 0.000000 0.8300000 0.17000000  
## Dead 0.000000 0.0000000 1.00000000  
##   
## , , Cycle74  
##   
## PFS OS Dead  
## PFS 0.8683351 0.1029738 0.02869112  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle75  
##   
## PFS OS Dead  
## PFS 0.8676979 0.1034758 0.02882632  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle76  
##   
## PFS OS Dead  
## PFS 0.8670665 0.1039732 0.02896034  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle77  
##   
## PFS OS Dead  
## PFS 0.8664407 0.1044661 0.0290932  
## OS 0.0000000 0.8300000 0.1700000  
## Dead 0.0000000 0.0000000 1.0000000  
##   
## , , Cycle78  
##   
## PFS OS Dead  
## PFS 0.8658203 0.1049548 0.02922492  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle79  
##   
## PFS OS Dead  
## PFS 0.8652052 0.1054392 0.02935553  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle80  
##   
## PFS OS Dead  
## PFS 0.8645954 0.1059195 0.02948505  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle81  
##   
## PFS OS Dead  
## PFS 0.8639908 0.1063957 0.0296135  
## OS 0.0000000 0.8300000 0.1700000  
## Dead 0.0000000 0.0000000 1.0000000  
##   
## , , Cycle82  
##   
## PFS OS Dead  
## PFS 0.8633911 0.106868 0.02974091  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle83  
##   
## PFS OS Dead  
## PFS 0.8627964 0.1073363 0.02986729  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle84  
##   
## PFS OS Dead  
## PFS 0.8622065 0.1078008 0.02999267  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle85  
##   
## PFS OS Dead  
## PFS 0.8616214 0.1082616 0.03011707  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle86  
##   
## PFS OS Dead  
## PFS 0.8610408 0.1087187 0.0302405  
## OS 0.0000000 0.8300000 0.1700000  
## Dead 0.0000000 0.0000000 1.0000000  
##   
## , , Cycle87  
##   
## PFS OS Dead  
## PFS 0.8604649 0.1091721 0.03036299  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle88  
##   
## PFS OS Dead  
## PFS 0.8598934 0.1096221 0.03048454  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle89  
##   
## PFS OS Dead  
## PFS 0.8593262 0.1100686 0.03060519  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle90  
##   
## PFS OS Dead  
## PFS 0.8587634 0.1105116 0.03072494  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle91  
##   
## PFS OS Dead  
## PFS 0.8582048 0.1109514 0.03084382  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle92  
##   
## PFS OS Dead  
## PFS 0.8576504 0.1113878 0.03096183  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle93  
##   
## PFS OS Dead  
## PFS 0.8571 0.111821 0.031079  
## OS 0.0000 0.830000 0.170000  
## Dead 0.0000 0.000000 1.000000  
##   
## , , Cycle94  
##   
## PFS OS Dead  
## PFS 0.8565536 0.112251 0.03119534  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle95  
##   
## PFS OS Dead  
## PFS 0.8560112 0.112678 0.03131086  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle96  
##   
## PFS OS Dead  
## PFS 0.8554726 0.1131018 0.03142558  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle97  
##   
## PFS OS Dead  
## PFS 0.8549378 0.1135227 0.03153951  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle98  
##   
## PFS OS Dead  
## PFS 0.8544067 0.1139406 0.03165267  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle99  
##   
## PFS OS Dead  
## PFS 0.8538794 0.1143556 0.03176507  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle100  
##   
## PFS OS Dead  
## PFS 0.8533556 0.1147677 0.03187671  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle101  
##   
## PFS OS Dead  
## PFS 0.8528354 0.115177 0.03198762  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle102  
##   
## PFS OS Dead  
## PFS 0.8523186 0.1155836 0.03209781  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle103  
##   
## PFS OS Dead  
## PFS 0.8518053 0.1159874 0.03220728  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle104  
##   
## PFS OS Dead  
## PFS 0.8512954 0.1163886 0.03231606  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle105  
##   
## PFS OS Dead  
## PFS 0.8507888 0.1167871 0.03242414  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle106  
##   
## PFS OS Dead  
## PFS 0.8502855 0.117183 0.03253154  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle107  
##   
## PFS OS Dead  
## PFS 0.8497854 0.1175764 0.03263828  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle108  
##   
## PFS OS Dead  
## PFS 0.8492884 0.1179672 0.03274435  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle109  
##   
## PFS OS Dead  
## PFS 0.8487946 0.1183556 0.03284978  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle110  
##   
## PFS OS Dead  
## PFS 0.8483039 0.1187415 0.03295457  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle111  
##   
## PFS OS Dead  
## PFS 0.8478162 0.119125 0.03305874  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle112  
##   
## PFS OS Dead  
## PFS 0.8473315 0.1195062 0.03316228  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle113  
##   
## PFS OS Dead  
## PFS 0.8468498 0.119885 0.03326521  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle114  
##   
## PFS OS Dead  
## PFS 0.8463709 0.1202616 0.03336755  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle115  
##   
## PFS OS Dead  
## PFS 0.8458949 0.1206358 0.03346928  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle116  
##   
## PFS OS Dead  
## PFS 0.8454217 0.1210078 0.03357044  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle117  
##   
## PFS OS Dead  
## PFS 0.8449513 0.1213777 0.03367102  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle118  
##   
## PFS OS Dead  
## PFS 0.8444836 0.1217453 0.03377103  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle119  
##   
## PFS OS Dead  
## PFS 0.8440187 0.1221108 0.03387049  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle120  
##   
## PFS OS Dead  
## PFS 0.8435564 0.1224742 0.03396939  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle121  
##   
## PFS OS Dead  
## PFS 0.8430967 0.1228356 0.03406774  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle122  
##   
## PFS OS Dead  
## PFS 0.8426396 0.1231948 0.03416557  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle123  
##   
## PFS OS Dead  
## PFS 0.8421851 0.1235521 0.03426286  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle124  
##   
## PFS OS Dead  
## PFS 0.8417331 0.1239073 0.03435963  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle125  
##   
## PFS OS Dead  
## PFS 0.8412835 0.1242606 0.03445588  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle126  
##   
## PFS OS Dead  
## PFS 0.8408365 0.1246119 0.03455163  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle127  
##   
## PFS OS Dead  
## PFS 0.8403918 0.1249613 0.03464687  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle128  
##   
## PFS OS Dead  
## PFS 0.8399496 0.1253088 0.03474162  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle129  
##   
## PFS OS Dead  
## PFS 0.8395097 0.1256544 0.03483588  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle130  
##   
## PFS OS Dead  
## PFS 0.8390721 0.1259982 0.03492965  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle131  
##   
## PFS OS Dead  
## PFS 0.8386369 0.1263402 0.03502296  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle132  
##   
## PFS OS Dead  
## PFS 0.8382039 0.1266803 0.03511579  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle133  
##   
## PFS OS Dead  
## PFS 0.8377731 0.1270187 0.03520815  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle134  
##   
## PFS OS Dead  
## PFS 0.8373446 0.1273553 0.03530006  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle135  
##   
## PFS OS Dead  
## PFS 0.8369183 0.1276902 0.03539151  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle136  
##   
## PFS OS Dead  
## PFS 0.8364941 0.1280234 0.03548252  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle137  
##   
## PFS OS Dead  
## PFS 0.8360721 0.1283549 0.03557309  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle138  
##   
## PFS OS Dead  
## PFS 0.8356521 0.1286847 0.03566321  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle139  
##   
## PFS OS Dead  
## PFS 0.8352343 0.1290128 0.03575291  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle140  
##   
## PFS OS Dead  
## PFS 0.8348185 0.1293393 0.03584218  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle141  
##   
## PFS OS Dead  
## PFS 0.8344048 0.1296642 0.03593103  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle142  
##   
## PFS OS Dead  
## PFS 0.833993 0.1299875 0.03601946  
## OS 0.000000 0.8300000 0.17000000  
## Dead 0.000000 0.0000000 1.00000000  
##   
## , , Cycle143  
##   
## PFS OS Dead  
## PFS 0.8335833 0.1303092 0.03610748  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000

# Using the transition probabilities for standard of care as basis, update the transition probabilities that are different for the experimental strategy  
  
#   
# m\_P\_Exp["PFS", "PFS", ] <- p\_PFS\_Exp  
# m\_P\_Exp["PFS", "AE1", ] <- p\_FA1\_Exp  
# m\_P\_Exp["PFS", "AE2", ] <- p\_FA2\_Exp  
# m\_P\_Exp["PFS", "AE3", ] <- p\_FA3\_Exp  
# m\_P\_Exp["PFS", "OS", ] <- p\_PFS\_OS\_Exp  
# m\_P\_Exp["PFS", "Dead", ] <- p\_PFSD\_Exp  
#   
#   
# # Setting the transition probabilities from OS  
# m\_P\_Exp["OS", "OS", ] <- 1 - P\_OSD\_Exp  
# m\_P\_Exp["OS", "Dead", ] <- P\_OSD\_Exp  
#   
# # Setting the transition probabilities from Dead  
# m\_P\_Exp["Dead", "Dead", ] <- 1  
#   
  
# Setting the transition probabilities the experimental strategy  
m\_P\_Exp["PFS", "PFS",]<- (1 -p\_PFSOS\_Exp) \* (1 - p\_PFSD\_Exp)  
m\_P\_Exp["PFS", "OS",]<- p\_PFSOS\_Exp\*(1 - p\_PFSD\_Exp)  
m\_P\_Exp["PFS", "Dead",]<-p\_PFSD\_Exp  
  
# Setting the transition probabilities from OS  
m\_P\_Exp["OS", "OS", ] <- 1 - P\_OSD\_Exp  
m\_P\_Exp["OS", "Dead", ] <- P\_OSD\_Exp  
  
# Setting the transition probabilities from Dead  
m\_P\_Exp["Dead", "Dead", ] <- 1  
  
  
  
  
  
  
# # Setting the transition probabilities from AE1  
# m\_P\_Exp["AE1", "PFS", ] <- p\_A1F\_Exp  
# m\_P\_Exp["AE1", "Dead", ] <- p\_A1D\_Exp  
#   
# # Setting the transition probabilities from AE2  
# m\_P\_Exp["AE2", "PFS", ] <- p\_A2F\_Exp  
# m\_P\_Exp["AE2", "Dead", ] <- p\_A2D\_Exp  
#   
# # Setting the transition probabilities from AE3  
# m\_P\_Exp["AE3", "PFS", ] <- p\_A3F\_Exp  
# m\_P\_Exp["AE3", "Dead", ] <- p\_A3D\_Exp  
#   
m\_P\_Exp

## , , Cycle1  
##   
## PFS OS Dead  
## PFS 0.9868477 0.01008927 0.003063016  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle2  
##   
## PFS OS Dead  
## PFS 0.9789383 0.01630433 0.004757328  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle3  
##   
## PFS OS Dead  
## PFS 0.9743709 0.01991833 0.005710751  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle4  
##   
## PFS OS Dead  
## PFS 0.9708731 0.02269335 0.006433555  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle5  
##   
## PFS OS Dead  
## PFS 0.9679654 0.02500413 0.007030492  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle6  
##   
## PFS OS Dead  
## PFS 0.965443 0.02701112 0.007545832  
## OS 0.000000 0.83000000 0.170000000  
## Dead 0.000000 0.00000000 1.000000000  
##   
## , , Cycle7  
##   
## PFS OS Dead  
## PFS 0.9631965 0.02880036 0.008003102  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle8  
##   
## PFS OS Dead  
## PFS 0.9611593 0.03042417 0.008416512  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle9  
##   
## PFS OS Dead  
## PFS 0.9592875 0.03191712 0.008795392  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle10  
##   
## PFS OS Dead  
## PFS 0.9575504 0.03330337 0.009146241  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle11  
##   
## PFS OS Dead  
## PFS 0.9559256 0.03460062 0.00947379  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle12  
##   
## PFS OS Dead  
## PFS 0.9543962 0.03582224 0.009781605  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle13  
##   
## PFS OS Dead  
## PFS 0.9529489 0.03697862 0.01007245  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle14  
##   
## PFS OS Dead  
## PFS 0.9515735 0.03807803 0.01034851  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle15  
##   
## PFS OS Dead  
## PFS 0.9502613 0.03912715 0.01061156  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle16  
##   
## PFS OS Dead  
## PFS 0.9490055 0.04013149 0.01086304  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle17  
##   
## PFS OS Dead  
## PFS 0.9478002 0.04109565 0.01110417  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle18  
##   
## PFS OS Dead  
## PFS 0.9466406 0.04202349 0.01133595  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle19  
##   
## PFS OS Dead  
## PFS 0.9455224 0.04291833 0.01155927  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle20  
##   
## PFS OS Dead  
## PFS 0.9444421 0.043783 0.01177485  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle21  
##   
## PFS OS Dead  
## PFS 0.9433967 0.04461998 0.01198335  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle22  
##   
## PFS OS Dead  
## PFS 0.9423833 0.04543142 0.01218532  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle23  
##   
## PFS OS Dead  
## PFS 0.9413995 0.04621922 0.01238126  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle24  
##   
## PFS OS Dead  
## PFS 0.9404433 0.04698506 0.01257161  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle25  
##   
## PFS OS Dead  
## PFS 0.9395128 0.04773043 0.01275675  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle26  
##   
## PFS OS Dead  
## PFS 0.9386063 0.04845668 0.01293702  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle27  
##   
## PFS OS Dead  
## PFS 0.9377223 0.049165 0.01311275  
## OS 0.0000000 0.830000 0.17000000  
## Dead 0.0000000 0.000000 1.00000000  
##   
## , , Cycle28  
##   
## PFS OS Dead  
## PFS 0.9368593 0.04985647 0.01328421  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle29  
##   
## PFS OS Dead  
## PFS 0.9360163 0.05053208 0.01345165  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle30  
##   
## PFS OS Dead  
## PFS 0.935192 0.05119273 0.0136153  
## OS 0.000000 0.83000000 0.1700000  
## Dead 0.000000 0.00000000 1.0000000  
##   
## , , Cycle31  
##   
## PFS OS Dead  
## PFS 0.9343854 0.05183922 0.01377537  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle32  
##   
## PFS OS Dead  
## PFS 0.9335956 0.05247229 0.01393206  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle33  
##   
## PFS OS Dead  
## PFS 0.9328218 0.05309265 0.01408554  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle34  
##   
## PFS OS Dead  
## PFS 0.9320631 0.0537009 0.01423596  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle35  
##   
## PFS OS Dead  
## PFS 0.9313189 0.05429764 0.01438349  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle36  
##   
## PFS OS Dead  
## PFS 0.9305883 0.0548834 0.01452825  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle37  
##   
## PFS OS Dead  
## PFS 0.9298709 0.05545867 0.01467038  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle38  
##   
## PFS OS Dead  
## PFS 0.9291661 0.05602392 0.01480999  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle39  
##   
## PFS OS Dead  
## PFS 0.9284732 0.05657958 0.0149472  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle40  
##   
## PFS OS Dead  
## PFS 0.9277919 0.05712606 0.01508209  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle41  
##   
## PFS OS Dead  
## PFS 0.9271215 0.05766371 0.01521478  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle42  
##   
## PFS OS Dead  
## PFS 0.9264618 0.0581929 0.01534535  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle43  
##   
## PFS OS Dead  
## PFS 0.9258122 0.05871395 0.01547388  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle44  
##   
## PFS OS Dead  
## PFS 0.9251724 0.05922718 0.01560045  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle45  
##   
## PFS OS Dead  
## PFS 0.924542 0.05973286 0.01572513  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle46  
##   
## PFS OS Dead  
## PFS 0.9239207 0.06023128 0.015848  
## OS 0.0000000 0.83000000 0.170000  
## Dead 0.0000000 0.00000000 1.000000  
##   
## , , Cycle47  
##   
## PFS OS Dead  
## PFS 0.9233082 0.0607227 0.01596913  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle48  
##   
## PFS OS Dead  
## PFS 0.9227041 0.06120734 0.01608856  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle49  
##   
## PFS OS Dead  
## PFS 0.9221082 0.06168546 0.01620636  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle50  
##   
## PFS OS Dead  
## PFS 0.9215202 0.06215725 0.01632259  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle51  
##   
## PFS OS Dead  
## PFS 0.9209398 0.06262293 0.01643729  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle52  
##   
## PFS OS Dead  
## PFS 0.9203668 0.0630827 0.01655052  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle53  
##   
## PFS OS Dead  
## PFS 0.9198009 0.06353673 0.01666233  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle54  
##   
## PFS OS Dead  
## PFS 0.919242 0.06398521 0.01677275  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle55  
##   
## PFS OS Dead  
## PFS 0.9186899 0.0644283 0.01688183  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle56  
##   
## PFS OS Dead  
## PFS 0.9181442 0.06486617 0.01698961  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle57  
##   
## PFS OS Dead  
## PFS 0.9176049 0.06529895 0.01709614  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle58  
##   
## PFS OS Dead  
## PFS 0.9170718 0.06572681 0.01720143  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle59  
##   
## PFS OS Dead  
## PFS 0.9165446 0.06614988 0.01730554  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle60  
##   
## PFS OS Dead  
## PFS 0.9160232 0.06656828 0.01740849  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle61  
##   
## PFS OS Dead  
## PFS 0.9155075 0.06698216 0.01751032  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle62  
##   
## PFS OS Dead  
## PFS 0.9149973 0.06739162 0.01761106  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle63  
##   
## PFS OS Dead  
## PFS 0.9144925 0.06779678 0.01771073  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle64  
##   
## PFS OS Dead  
## PFS 0.9139929 0.06819776 0.01780936  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle65  
##   
## PFS OS Dead  
## PFS 0.9134984 0.06859466 0.01790699  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle66  
##   
## PFS OS Dead  
## PFS 0.9130088 0.06898759 0.01800363  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle67  
##   
## PFS OS Dead  
## PFS 0.9125241 0.06937664 0.01809931  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle68  
##   
## PFS OS Dead  
## PFS 0.912044 0.0697619 0.01819405  
## OS 0.000000 0.8300000 0.17000000  
## Dead 0.000000 0.0000000 1.00000000  
##   
## , , Cycle69  
##   
## PFS OS Dead  
## PFS 0.9115686 0.07014347 0.01828788  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle70  
##   
## PFS OS Dead  
## PFS 0.9110977 0.07052144 0.01838083  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle71  
##   
## PFS OS Dead  
## PFS 0.9106312 0.07089588 0.0184729  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle72  
##   
## PFS OS Dead  
## PFS 0.910169 0.07126689 0.01856412  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle73  
##   
## PFS OS Dead  
## PFS 0.909711 0.07163453 0.01865451  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle74  
##   
## PFS OS Dead  
## PFS 0.909257 0.07199888 0.01874409  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle75  
##   
## PFS OS Dead  
## PFS 0.9088071 0.07236002 0.01883288  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle76  
##   
## PFS OS Dead  
## PFS 0.9083611 0.07271801 0.01892089  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle77  
##   
## PFS OS Dead  
## PFS 0.9079189 0.07307292 0.01900814  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle78  
##   
## PFS OS Dead  
## PFS 0.9074805 0.07342483 0.01909465  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle79  
##   
## PFS OS Dead  
## PFS 0.9070458 0.07377378 0.01918044  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle80  
##   
## PFS OS Dead  
## PFS 0.9066147 0.07411984 0.01926551  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle81  
##   
## PFS OS Dead  
## PFS 0.906187 0.07446307 0.01934988  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle82  
##   
## PFS OS Dead  
## PFS 0.9057629 0.07480353 0.01943358  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle83  
##   
## PFS OS Dead  
## PFS 0.9053421 0.07514127 0.0195166  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle84  
##   
## PFS OS Dead  
## PFS 0.9049247 0.07547635 0.01959897  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle85  
##   
## PFS OS Dead  
## PFS 0.9045105 0.07580881 0.01968069  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle86  
##   
## PFS OS Dead  
## PFS 0.9040995 0.07613871 0.01976179  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle87  
##   
## PFS OS Dead  
## PFS 0.9036916 0.0764661 0.01984227  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle88  
##   
## PFS OS Dead  
## PFS 0.9032868 0.07679102 0.01992214  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle89  
##   
## PFS OS Dead  
## PFS 0.9028851 0.07711352 0.02000141  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle90  
##   
## PFS OS Dead  
## PFS 0.9024863 0.07743364 0.02008011  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle91  
##   
## PFS OS Dead  
## PFS 0.9020903 0.07775143 0.02015823  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle92  
##   
## PFS OS Dead  
## PFS 0.9016973 0.07806692 0.02023578  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle93  
##   
## PFS OS Dead  
## PFS 0.901307 0.07838017 0.02031279  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle94  
##   
## PFS OS Dead  
## PFS 0.9009196 0.0786912 0.02038925  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle95  
##   
## PFS OS Dead  
## PFS 0.9005348 0.07900006 0.02046518  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle96  
##   
## PFS OS Dead  
## PFS 0.9001526 0.07930678 0.02054058  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle97  
##   
## PFS OS Dead  
## PFS 0.8997731 0.0796114 0.02061547  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle98  
##   
## PFS OS Dead  
## PFS 0.8993962 0.07991396 0.02068985  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle99  
##   
## PFS OS Dead  
## PFS 0.8990218 0.08021449 0.02076374  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle100  
##   
## PFS OS Dead  
## PFS 0.8986498 0.08051302 0.02083714  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle101  
##   
## PFS OS Dead  
## PFS 0.8982804 0.08080959 0.02091005  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle102  
##   
## PFS OS Dead  
## PFS 0.8979133 0.08110422 0.02098249  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle103  
##   
## PFS OS Dead  
## PFS 0.8975486 0.08139695 0.02105447  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle104  
##   
## PFS OS Dead  
## PFS 0.8971862 0.08168781 0.02112599  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle105  
##   
## PFS OS Dead  
## PFS 0.8968261 0.08197683 0.02119706  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle106  
##   
## PFS OS Dead  
## PFS 0.8964683 0.08226404 0.02126768  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle107  
##   
## PFS OS Dead  
## PFS 0.8961127 0.08254946 0.02133787  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle108  
##   
## PFS OS Dead  
## PFS 0.8957593 0.08283312 0.02140763  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle109  
##   
## PFS OS Dead  
## PFS 0.895408 0.08311505 0.02147696  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle110  
##   
## PFS OS Dead  
## PFS 0.8950589 0.08339527 0.02154587  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle111  
##   
## PFS OS Dead  
## PFS 0.8947118 0.08367382 0.02161438  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle112  
##   
## PFS OS Dead  
## PFS 0.8943668 0.0839507 0.02168248  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle113  
##   
## PFS OS Dead  
## PFS 0.8940239 0.08422596 0.02175018  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle114  
##   
## PFS OS Dead  
## PFS 0.8936829 0.08449961 0.02181749  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle115  
##   
## PFS OS Dead  
## PFS 0.8933439 0.08477167 0.02188441  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle116  
##   
## PFS OS Dead  
## PFS 0.8930069 0.08504217 0.02195095  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle117  
##   
## PFS OS Dead  
## PFS 0.8926718 0.08531113 0.02201712  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle118  
##   
## PFS OS Dead  
## PFS 0.8923385 0.08557857 0.02208291  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle119  
##   
## PFS OS Dead  
## PFS 0.8920072 0.08584451 0.02214834  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle120  
##   
## PFS OS Dead  
## PFS 0.8916776 0.08610897 0.02221341  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle121  
##   
## PFS OS Dead  
## PFS 0.8913499 0.08637198 0.02227812  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle122  
##   
## PFS OS Dead  
## PFS 0.891024 0.08663355 0.02234248  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle123  
##   
## PFS OS Dead  
## PFS 0.8906998 0.0868937 0.02240649  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle124  
##   
## PFS OS Dead  
## PFS 0.8903774 0.08715245 0.02247017  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle125  
##   
## PFS OS Dead  
## PFS 0.8900567 0.08740982 0.0225335  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle126  
##   
## PFS OS Dead  
## PFS 0.8897377 0.08766583 0.02259651  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle127  
##   
## PFS OS Dead  
## PFS 0.8894203 0.08792049 0.02265918  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle128  
##   
## PFS OS Dead  
## PFS 0.8891046 0.08817382 0.02272154  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle129  
##   
## PFS OS Dead  
## PFS 0.8887906 0.08842584 0.02278357  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle130  
##   
## PFS OS Dead  
## PFS 0.8884781 0.08867657 0.02284529  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle131  
##   
## PFS OS Dead  
## PFS 0.8881673 0.08892601 0.02290669  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle132  
##   
## PFS OS Dead  
## PFS 0.887858 0.0891742 0.02296779  
## OS 0.000000 0.8300000 0.17000000  
## Dead 0.000000 0.0000000 1.00000000  
##   
## , , Cycle133  
##   
## PFS OS Dead  
## PFS 0.8875503 0.08942114 0.02302859  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle134  
##   
## PFS OS Dead  
## PFS 0.8872441 0.08966684 0.02308908  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle135  
##   
## PFS OS Dead  
## PFS 0.8869394 0.08991133 0.02314928  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle136  
##   
## PFS OS Dead  
## PFS 0.8866362 0.09015462 0.02320919  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle137  
##   
## PFS OS Dead  
## PFS 0.8863345 0.09039672 0.0232688  
## OS 0.0000000 0.83000000 0.1700000  
## Dead 0.0000000 0.00000000 1.0000000  
##   
## , , Cycle138  
##   
## PFS OS Dead  
## PFS 0.8860342 0.09063765 0.02332813  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle139  
##   
## PFS OS Dead  
## PFS 0.8857354 0.09087741 0.02338718  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000  
##   
## , , Cycle140  
##   
## PFS OS Dead  
## PFS 0.885438 0.09111604 0.02344596  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle141  
##   
## PFS OS Dead  
## PFS 0.885142 0.09135353 0.02350445  
## OS 0.000000 0.83000000 0.17000000  
## Dead 0.000000 0.00000000 1.00000000  
##   
## , , Cycle142  
##   
## PFS OS Dead  
## PFS 0.8848474 0.0915899 0.02356267  
## OS 0.0000000 0.8300000 0.17000000  
## Dead 0.0000000 0.0000000 1.00000000  
##   
## , , Cycle143  
##   
## PFS OS Dead  
## PFS 0.8845542 0.09182516 0.02362063  
## OS 0.0000000 0.83000000 0.17000000  
## Dead 0.0000000 0.00000000 1.00000000

# If I wanted to round my transition matrix so things sum exactly to 1, instead of 0.99999999 (in cases where this was happening):  
#round(m\_P\_SoC, digits=2)   
#m\_P\_SoC

# An explanation of conditional probabilities:  
  
## Standard of Care  
# from PFS  
# m\_P\_SoC["PFS", "PFS"] <- (1 - p\_HD) \* (1 - p\_HS\_SoC)  
  
# It's 1- p\_HD, because if you're not going from healthy to dead then you're staying in healthy (i.e. you're not going from PFS to Dead, so you're staying in PFS) so this captures all the people leftover in PFS after those going to dead, and again for 1- p\_HS\_SoC, if you're not going from healthy to sick, the only other way out of healthy per the transition probability matrix, then you're staying in PFS, so 1- p\_HS\_SoC takes away all the people who went from healthy to sick, and leaves behind all the people who stayed in healthy (or takes away all the people who went to OS and leaves behind all the people that went to PFS per my model).  
  
# m\_P\_SoC["PFS", "OS"] <- (1 - p\_HD) \* p\_HS\_SoC  
  
# This is because, when setting up the input parameters we didn't have a value for staying in progression free survival, so we have to calculate it, i.e. above we have to get the probability for going from PFS to OS, by getting all the people left in PFS after the people who went to death were gone, and then we multiply this probability by the probability of going from healthy to sick, or PFS to OS, because it's the probability conditional on being alive, the probability conditional on being left in the PFS state after the other people who went to the death state are gone, so the probability of going from PFS to OS, conditional on surviving.  
  
  
  
# m\_P\_SoC["PFS", "OS"] <- (1 - p\_HD) \* p\_HS\_SoC  
  
# p\_HS\_SoC is the probability of transitioning from healthy to sick, conditional on surviving, so it's defined as a conditional probability. So, what the Markov model wants is : m\_P\_SoC["PFS", "OS"] what's the overall probability of transitioning from PFS to OS, and so that is actually not a conditional probability, which is why we multiply the probability of surviving (1 - p\_HD) \* p\_HS\_SoC by the probability of going to OS conditional on surviving, because the transition in the model should be the marginal not the conditional, i.e. you want to have the end probability.  
  
  
#So, because you took a probability that was conditional from the literature, or digitised a PFS curve to get a transition probability - and those probabilities come from people who were necessarily still in the PFS state when you digitise a PFS curve, so those are transition probabilities conditional on being in the PFS (or healthy) state - you need to do the multiplication to give you a probability you can put in your transition matrix that reflects that the probability you are working with is conditional.  
  
  
# Adverse event transition probabilities have all been conditional, this is to reflect that probabilities on transitioning from adverse event to death will necessarily be conditional probabilities - as they come from individuals who were studied in the adverse event group - so they are necessarily probabilities conditional on experiencing the adverse event. Likewise, the probability of experiencing the adverse event when under treatment is conditional on being in PFS for the treatment in the first place, because it will come from reports of adverse events in studies of the treatments given that we look at.   
  
  
# Which means conditional probabilities kind of don't matter too much, because the probability won't be applied to anyone who isnt in the state it doesnt necessarily need to be conditional after all. And I guess it would have never been applied to anyone who wasnt in the state, because probabilities are only applied to people who are in the state when the transition probability matrix is multiplied by the cohort trace of people in the state.  
  
  
  
  
# m\_P\_SoC["PFS", "Dead"] <- p\_HD  
  
# Your probability of going from healthy to dead is not conditional on surviving, per the input parameters section.  
  
  
# from OS  
# m\_P\_SoC["OS", "OS"] <- 1 - p\_SD  
# m\_P\_SoC["OS", "Dead"] <- p\_SD  
  
# Per the input parameters, your probability of going from sick to dead is also not conditional on surviving, so we dont need to multiply our probability conditional on surviving by the number of survivors as above.  
  
# That's why though, our PFS to OS probability calculated through multiplication above is so close to the 0.05 in the input parameters, i.e. it's 0.0495, because we multiply it by 1 - p\_HD or 1-0.01, which is basically multiplying it by a number very close to 1, so the probability remains very close to where it started off. The probability is conditional on the numbers of people in the alive state, we calculate that to be nearly 100% of people, so the probability remains the same.   
  
  
# m\_P\_SoC["PFS", "PFS"] <- (1 - p\_HD) \* (1 - p\_HS\_SoC)  
# And the first one (commented out above) is the same, it's the probability of going from healthy to healthy which, because it's a conditional probability, is the number of individuals left in healthy multiplied by the probability of going from healthy to health conditional on surviving to be in healthy (so, to get a conditional probability we multiply the numbers who have survived by the probability -> here we don't have a probability for healthy to healthy, but we do have a probability for health to sick, so taking 1 - this gives us the probability of not going into sick, i.e. of staying in healthy instead).   
  
# from Dead  
# m\_P\_SoC["Dead", "Dead"] <- 1  
  
# Once you're in dead you stay in dead, you can't come back to life.  
  
  
# For handiness, we just take the treatment matrix that already exists for standard of care above, copy it as treatment A and treatment B, and then copy over it with new values specific to treatment A and treatment B, assuming that the probability of going from PFS to dead and staying in OS and going from OS to dead are the same as for standard of care (dead to dead is definitely the same as no one leaves dead) that's likely a reasonable assumption for our model, as once people progress we can assume they all go on to the same next treatment, standard of care progression treatment with the same likelihood of staying in OS and of going to the dead state, regardless of the novel therapy they were on:  
  
# ## EPI Assay  
# m\_P\_trtA <- m\_P\_SoC  
# m\_P\_trtA["PFS", "PFS"] <- (1 - p\_HD) \* (1 - p\_HS\_trtA)  
# m\_P\_trtA["PFS", "OS"] <- (1 - p\_HD) \* p\_HS\_trtA  
#   
# ## HDX Assay  
# m\_P\_trtB <- m\_P\_SoC  
# m\_P\_trtB["PFS", "PFS"] <- (1 - p\_HD) \* (1 - p\_HS\_trtB)  
# m\_P\_trtB["PFS", "OS"] <- (1 - p\_HD) \* p\_HS\_trtB  
  
  
  
# You can also think of it, per the 1hour: 05minute mark of: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_24\Live Session Recording\Live Session Recording August 24th.mp4  
  
  
# m\_P\_SoC["PFS", "OS"] <- (1 - p\_HD) [<- if you don't die] \* p\_HS\_SoC [then you have a risk of getting sick]  
  
# It wont always be necessary to do this, i.e., to include conditional probabilities, it will depend on the data used and how things were estimated.  
  
# This approach is taken for conditional probabilities, it's probably more relevant for modelling people to really, really old ages, where their mortality gets really high (so this is particularly relevant when you have age specific mortality), where you still have this high mortality of getting sick, you can end up in problems with the probabilities where the probability of leaving the state gets too high, because you haven't properly adjusted for these really high mortality rates, as people get really old.  
  
# So, what this is doing is saying, if you don't die, then it's the remaining people who experience the probability of getting sick, so you're not adding probabilities together, which can be problematic, because sometimes you can have probabilities come to greater than one (i.e., larger than 100%).  
  
# It's always worth considering, if you're extrapolating this risk of illness that's pretty high, on top of a cohort that also has a high risk of death, maybe that's not the right extrapolation, but this is another kind of fail safe to avoid that.  
  
# Depending on how the data is estimated, this a correct interpretation of the probability estimate of going from one state to another (p\_HS\_SoC, or healthy to sick, or PFS to OS).   
  
  
# Using conditional probabilities more generally is also discussed around the 1:04 hour mark of:  
  
# C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \\_ DARTH\August\\_25\Live Session Recording\Live Session Recording August 25th WITH CHAT.mkv

I may want to get rid of the conditional probabilities above, replacing them with standard transition probabilities per my York training.

## 04.4 Check if transition probability matrices are valid.

# This is a check in the DARTH tools package that all the transition probabilities are in [0, 1], i.e., no probabilities are greater than 100%.  
# This works as follows, according to line 205 of: https://github.com/DARTH-git/cohort-modeling-tutorial-intro/blob/main/analysis/cSTM\_time\_indep.R  
#   
# ## Check if transition probability matrices are valid ----  
# #\* Functions included in "R/Functions.R". The latest version can be found in `darthtools` package  
# ### Check that transition probabilities are [0, 1] ----  
# check\_transition\_probability(m\_P, verbose = TRUE) # m\_P >= 0 && m\_P <= 1  
# check\_transition\_probability(m\_P\_strA, verbose = TRUE) # m\_P\_strA >= 0 && m\_P\_strA <= 1  
# check\_transition\_probability(m\_P\_strB, verbose = TRUE) # m\_P\_strB >= 0 && m\_P\_strB <= 1  
# check\_transition\_probability(m\_P\_strAB, verbose = TRUE) # m\_P\_strAB >= 0 && m\_P\_strAB <= 1  
# ### Check that all rows sum to 1 ----  
# check\_sum\_of\_transition\_array(m\_P, n\_states = n\_states, n\_cycles = n\_cycles, verbose = TRUE) # rowSums(m\_P) == 1  
# check\_sum\_of\_transition\_array(m\_P\_strA, n\_states = n\_states, n\_cycles = n\_cycles, verbose = TRUE) # rowSums(m\_P\_strA) == 1  
# check\_sum\_of\_transition\_array(m\_P\_strB, n\_states = n\_states, n\_cycles = n\_cycles, verbose = TRUE) # rowSums(m\_P\_strB) == 1  
# check\_sum\_of\_transition\_array(m\_P\_strAB, n\_states = n\_states, n\_cycles = n\_cycles, verbose = TRUE) # rowSums(m\_P\_strAB) == 1  
#   
  
  
  
  
# UNCOMMENT THIS!   
check\_transition\_probability(m\_P\_SoC, verbose = TRUE)  
# UNCOMMENT THIS!   
check\_transition\_probability(m\_P\_Exp, verbose = TRUE)  
#check\_transition\_probability(m\_P\_trtA, verbose = TRUE)  
#check\_transition\_probability(m\_P\_trtB, verbose = TRUE)  
# Check that all rows sum in each matrix sum to 1 -> which we know is a necessary condition for transition probability matrices.  
# UNCOMMENT THIS!   
check\_sum\_of\_transition\_array(m\_P\_SoC, n\_states = n\_states, n\_cycles = n\_cycle, verbose = TRUE)  
# UNCOMMENT THIS!   
check\_sum\_of\_transition\_array(m\_P\_Exp, n\_states = n\_states, n\_cycles = n\_cycle, verbose = TRUE)  
#check\_sum\_of\_transition\_array(m\_P\_trtA, n\_states = n\_states, n\_cycles = n\_cycle, verbose = TRUE)  
#check\_sum\_of\_transition\_array(m\_P\_trtB, n\_states = n\_states, n\_cycles = n\_cycle, verbose = TRUE)  
  
# This error message:  
#   
# Error in check\_sum\_of\_transition\_array(m\_P\_SoC, n\_states = n\_states, n\_cycles = n\_cycle, :   
# This is not a valid transition array   
#   
# Probably reflects that:   
#   
# > m\_P\_SoC  
# , , Cycle1  
#   
# PFS OS Dead AE1 AE2 AE3  
# PFS 0.974925 0.005074995 0.020000000 0.0194985 0.0194985 0.0194985  
# [1] 1.058495  
#   
# That is, that I've put just any old probability value in at the moment and they are summing to be larger than one, when I've changed this to the actual probabilities I'm sure it'll be OK.  
#   
# Same with:  
#   
# , , Cycle28  
#   
# PFS OS Dead AE1 AE2 AE3  
# PFS 0.5804491 0.3995509 0.020000 0.01160898 0.01160898 0.01160898  
# OS 0.0000000 0.5922950 0.407705 0.00000000 0.00000000 0.00000000  
# Dead 0.0000000 0.0000000 1.000000 0.00000000 0.00000000 0.00000000  
# AE1 0.9990000 0.0000000 0.001000 0.00000000 0.00000000 0.00000000   
#   
# > 0.5804491 + 0.3995509 + 0.020000 + 0.01160898 + 0.01160898 + 0.01160898  
# [1] 1.034827  
  
# Inspect whether properly defined  
# - note that we inspect for the first two cycles  
m\_P\_SoC[ , , 1:2]

## , , Cycle1  
##   
## PFS OS Dead  
## PFS 0.9805142 0.01477733 0.004708444  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle2  
##   
## PFS OS Dead  
## PFS 0.9688674 0.02382301 0.007309584  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000

m\_P\_Exp[ , , 1:2]

## , , Cycle1  
##   
## PFS OS Dead  
## PFS 0.9868477 0.01008927 0.003063016  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000  
##   
## , , Cycle2  
##   
## PFS OS Dead  
## PFS 0.9789383 0.01630433 0.004757328  
## OS 0.0000000 0.83000000 0.170000000  
## Dead 0.0000000 0.00000000 1.000000000

# Does it visually sum to 1?

# 05 Run Markov model

Looking in the below where more than 100% can be in a health state (1.02 in PFS in cycle 4), it seems likely that it’s the transitions from adverse events into PFS that could be causing the problems with my probabilities not summing to 1 as per verbose, as the only way to get into PFS is through one of the AE’s. I’ll focus on this possibility when repairing the transition probabilities.

PFS AE1 AE2 AE3 OS Dead

Cycle 4 1.0233086 0.01933076 0.01933076 0.01933076 0.031888685 0.06024724

# for (t in 1:n\_cycles){ # Use a for loop to loop through the number of cycles, basically we'll calculate the cohort distribution at the next cycle [t+1] based on the matrix of where they were at time t, matrix multiplied by the transition probability matrix for the current cycle (constant for us as we use a constant transition probability matrix, rather than a transition probability array).  
# We do this for each treatment, as they all have different transition probability matrices.   
# m\_M\_SoC [t + 1, ] <- m\_M\_SoC [t, ] %\*% m\_P\_SoC # estimate the state vector for the next cycle (t + 1)  
# m\_M\_trtA[t + 1, ] <- m\_M\_trtA[t, ] %\*% m\_P\_trtA # estimate the state vector for the next cycle (t + 1)  
# m\_M\_trtB[t + 1, ] <- m\_M\_trtB[t, ] %\*% m\_P\_trtB # estimate the state vector for the next cycle (t + 1)  
# }  
# head(m\_M\_SoC) # print the first few lines of the matrix for standard of care (m\_M\_SoC)  
  
  
# Above I was originally using a transition probability matrix, because I was using a time constant transition probability. But now, because my transition probabilities come from the survival curves, and thus change over time, I am using a transition probability array.   
  
# Thus, I adapt the above to reflect that my transition probability selected has to come from a certain cycle, i.e. from a certain time, and then be multiplied by the number of people in the matrix, the amount of the cohort in the matrix, at that cycle, i.e. at that time. Thats why below I pick the third dimension of the array, not row, not column, but time: R,C,T i.e. -> , ,i\_cycle  
  
  
  
# So here I once again create the Markov cohort trace by looping over all cycles  
# - note that the trace can easily be obtained using matrix multiplications  
# - note that now the right probabilities for the cycle need to be selected, like I explained above.  
for(i\_cycle in 1:(n\_cycle-1)) {  
 m\_M\_SoC[i\_cycle + 1, ] <- m\_M\_SoC[i\_cycle, ] %\*% m\_P\_SoC[ , , i\_cycle]  
 m\_M\_Exp[i\_cycle + 1, ] <- m\_M\_Exp[i\_cycle, ] %\*% m\_P\_Exp[ , , i\_cycle]  
}  
  
  
head(m\_M\_SoC) # print the first few lines of the matrix for standard of care (m\_M\_SoC)

## PFS OS Dead  
## Cycle 1 1.0000000 0.00000000 0.000000000  
## Cycle 2 0.9805142 0.01477733 0.004708444  
## Cycle 3 0.9499883 0.03562399 0.014387742  
## Cycle 4 0.9140449 0.05717774 0.028777356  
## Cycle 5 0.8747794 0.07769169 0.047528898  
## Cycle 6 0.8334820 0.09633774 0.070180295

head(m\_M\_Exp) # print the first few lines of the matrix for experimental treatment(m\_M\_Exp)

## PFS OS Dead  
## Cycle 1 1.0000000 0.00000000 0.000000000  
## Cycle 2 0.9868477 0.01008927 0.003063016  
## Cycle 3 0.9660631 0.02446398 0.009472949  
## Cycle 4 0.9413038 0.03954747 0.019148772  
## Cycle 5 0.9138865 0.05418573 0.031927771  
## Cycle 6 0.8846105 0.06782509 0.047564417

#m\_M\_SoC  
m\_M\_SoC

## PFS OS Dead  
## Cycle 1 1.000000000000000 0.00000000000000 0.000000000  
## Cycle 2 0.980514224355189 0.01477733151817 0.004708444  
## Cycle 3 0.949988269604735 0.03562398843649 0.014387742  
## Cycle 4 0.914044904041534 0.05717774013149 0.028777356  
## Cycle 5 0.874779409562102 0.07769169255661 0.047528898  
## Cycle 6 0.833481962065319 0.09633774303070 0.070180295  
## Cycle 7 0.791064941395003 0.11272115807205 0.096213901  
## Cycle 8 0.748215368460346 0.12668918735961 0.125095444  
## Cycle 9 0.705466779581530 0.13823434451161 0.156298876  
## Cycle 10 0.663239294608142 0.14743869243752 0.189322013  
## Cycle 11 0.621864731118115 0.15443915851055 0.223696110  
## Cycle 12 0.581603788716319 0.15940487909169 0.258991332  
## Cycle 13 0.542658625121295 0.16252196095655 0.294819414  
## Cycle 14 0.505182546602109 0.16398306671547 0.330834387  
## Cycle 15 0.469287772027293 0.16398026217220 0.366731966  
## Cycle 16 0.435051836561959 0.16270013323240 0.402248030  
## Cycle 17 0.402522985942753 0.16032051508845 0.437156499  
## Cycle 18 0.371724788765310 0.15700838369547 0.471266828  
## Cycle 19 0.342660120443168 0.15291859313209 0.504421286  
## Cycle 20 0.315314626920171 0.14819323148619 0.536492142  
## Cycle 21 0.289659747226042 0.14296142897686 0.567378824  
## Cycle 22 0.265655355023687 0.13733949493323 0.597005150  
## Cycle 23 0.243252066600087 0.13143129103155 0.625316642  
## Cycle 24 0.222393254015667 0.12532877067729 0.652277975  
## Cycle 25 0.203016795943699 0.11911263110590 0.677870573  
## Cycle 26 0.185056594218429 0.11285303733588 0.702090368  
## Cycle 27 0.168443880703941 0.10661038667777 0.724945733  
## Cycle 28 0.153108336430133 0.10043608987540 0.746455574  
## Cycle 29 0.138979042778989 0.09437335068715 0.766647607  
## Cycle 30 0.125985282686769 0.08845793020793 0.785556787  
## Cycle 31 0.114057208252445 0.08271888577477 0.803223906  
## Cycle 32 0.103126389741685 0.07717927710692 0.819694333  
## Cycle 33 0.093126259704313 0.07185683456219 0.835016906  
## Cycle 34 0.083992464752460 0.06676458616704 0.849242949  
## Cycle 35 0.075663136458018 0.06191144148872 0.862425422  
## Cycle 36 0.068079091809629 0.05730273153562 0.874618177  
## Cycle 37 0.061183972714055 0.05294070475198 0.885875323  
## Cycle 38 0.054924333129729 0.04882497985839 0.896250687  
## Cycle 39 0.049249681578937 0.04495295681628 0.905797362  
## Cycle 40 0.044112485997444 0.04132018759004 0.914567326  
## Cycle 41 0.039468147145270 0.03792070866867 0.922611144  
## Cycle 42 0.035274946118543 0.03474733750811 0.929977716  
## Cycle 43 0.031493970868762 0.03179193518234 0.936714094  
## Cycle 44 0.028089026051177 0.02904563759849 0.942865336  
## Cycle 45 0.025026529986772 0.02649905765047 0.948474412  
## Cycle 46 0.022275402031035 0.02414246066583 0.953582137  
## Cycle 47 0.019806943195286 0.02196591544985 0.958227141  
## Cycle 48 0.017594712460925 0.01995942315664 0.962445864  
## Cycle 49 0.015614400861353 0.01811302612370 0.966272573  
## Cycle 50 0.013843705078200 0.01641689870036 0.969739396  
## Cycle 51 0.012262202005587 0.01486142198469 0.972876376  
## Cycle 52 0.010851225476030 0.01343724426187 0.975711530  
## Cycle 53 0.009593746111870 0.01213532881241 0.978270925  
## Cycle 54 0.008474255064446 0.01094699063308 0.980578754  
## Cycle 55 0.007478652227290 0.00986392348928 0.982657424  
## Cycle 56 0.006594139357118 0.00887821859636 0.984527642  
## Cycle 57 0.005809118405234 0.00798237610962 0.986208505  
## Cycle 58 0.005113095250009 0.00716931049040 0.987717594  
## Cycle 59 0.004496588926442 0.00643235070879 0.989071060  
## Cycle 60 0.003951046369569 0.00576523614244 0.990283717  
## Cycle 61 0.003468762622999 0.00516210893670 0.991369128  
## Cycle 62 0.003042806410445 0.00461750350327 0.992339690  
## Cycle 63 0.002666950925376 0.00412633375331 0.993206715  
## Cycle 64 0.002335609660493 0.00368387858622 0.993980512  
## Cycle 65 0.002043777073313 0.00328576608671 0.994670457  
## Cycle 66 0.001786973865720 0.00292795682084 0.995285069  
## Cycle 67 0.001561196642821 0.00260672656509 0.995832077  
## Cycle 68 0.001362871708969 0.00231864875212 0.996318480  
## Cycle 69 0.001188812755511 0.00206057687118 0.996750610  
## Cycle 70 0.001036182195098 0.00182962702103 0.997134191  
## Cycle 71 0.000902455900424 0.00162316077707 0.997474383  
## Cycle 72 0.000785391110697 0.00143876850308 0.997775840  
## Cycle 73 0.000682997276336 0.00127425321025 0.998042750  
## Cycle 74 0.000593509621029 0.00112761504204 0.998278875  
## Cycle 75 0.000515365209902 0.00099703644281 0.998487598  
## Cycle 76 0.000447181322964 0.00088086805020 0.998671951  
## Cycle 77 0.000387735943832 0.00077761533604 0.998834649  
## Cycle 78 0.000335950184822 0.00068592600788 0.998978124  
## Cycle 79 0.000290872480662 0.00060457817238 0.999104549  
## Cycle 80 0.000251664394166 0.00053246925326 0.999215866  
## Cycle 81 0.000217587888052 0.00046860564888 0.999313806  
## Cycle 82 0.000187993927664 0.00041209310938 0.999399913  
## Cycle 83 0.000162312289573 0.00036212780836 0.999475560  
## Cycle 84 0.000140042460746 0.00031798808100 0.999541969  
## Cycle 85 0.000120745522300 0.00027902679795 0.999600228  
## Cycle 86 0.000104036920640 0.00024466434276 0.999651299  
## Cycle 87 0.000089580037059 0.00021438215957 0.999696038  
## Cycle 88 0.000077080474650 0.00018771683752 0.999735203  
## Cycle 89 0.000066280988637 0.00016425469816 0.999769464  
## Cycle 90 0.000056956992976 0.00014362685296 0.999799416  
## Cycle 91 0.000048912582329 0.00012550469832 0.999825583  
## Cycle 92 0.000041977014311 0.00010959581686 0.999848427  
## Cycle 93 0.000036001602208 0.00009564025485 0.999868358  
## Cycle 94 0.000030856973290 0.00008340714663 0.999885736  
## Cycle 95 0.000026430652300 0.00007269165897 0.999900878  
## Cycle 96 0.000022624933798 0.00006331222899 0.999914063  
## Cycle 97 0.000019355010788 0.00005510807140 0.999925537  
## Cycle 98 0.000016547330443 0.00004793693201 0.999935516  
## Cycle 99 0.000014138150806 0.00004167306603 0.999944189  
## Cycle 100 0.000012072275186 0.00003620542114 0.999951722  
## Cycle 101 0.000010301943427 0.00003143600686 0.999958262  
## Cycle 102 0.000008785861555 0.00002727843288 0.999963936  
## Cycle 103 0.000007488353338 0.00002365660060 0.999968855  
## Cycle 104 0.000006378619129 0.00002050353319 0.999973118  
## Cycle 105 0.000005430089023 0.00001776033084 0.999976810  
## Cycle 106 0.000004619858848 0.00001537523881 0.999980005  
## Cycle 107 0.000003928198815 0.00001330281711 0.999982769  
## Cycle 108 0.000003338125850 0.00001150320152 0.999985159  
## Cycle 109 0.000002835031677 0.00000994144666 0.999987224  
## Cycle 110 0.000002406359673 0.00000858694255 0.999989007  
## Cycle 111 0.000002041324319 0.00000741289711 0.999990546  
## Cycle 112 0.000001730667866 0.00000639587745 0.999991873  
## Cycle 113 0.000001466449426 0.00000551540383 0.999993018  
## Cycle 114 0.000001241862337 0.00000475359052 0.999994005  
## Cycle 115 0.000001051076138 0.00000409482843 0.999994854  
## Cycle 116 0.000000889099938 0.00000352550503 0.999995585  
## Cycle 117 0.000000751664390 0.00000303375725 0.999996215  
## Cycle 118 0.000000635119807 0.00000260925379 0.999996756  
## Cycle 119 0.000000536348284 0.00000224300352 0.999997221  
## Cycle 120 0.000000452687965 0.00000192718686 0.999997620  
## Cycle 121 0.000000381867814 0.00000165500771 0.999997963  
## Cycle 122 0.000000321951489 0.00000142056335 0.999998257  
## Cycle 123 0.000000271289074 0.00000121873034 0.999998510  
## Cycle 124 0.000000228475608 0.00000104506451 0.999998726  
## Cycle 125 0.000000192315473 0.00000089571334 0.999998912  
## Cycle 126 0.000000161791841 0.00000076733931 0.999999071  
## Cycle 127 0.000000136040481 0.00000065705282 0.999999207  
## Cycle 128 0.000000114327308 0.00000056235363 0.999999323  
## Cycle 129 0.000000096029174 0.00000048107973 0.999999423  
## Cycle 130 0.000000080617422 0.00000041136267 0.999999508  
## Cycle 131 0.000000067643831 0.00000035158867 0.999999581  
## Cycle 132 0.000000056728611 0.00000030036473 0.999999643  
## Cycle 133 0.000000047550142 0.00000025648912 0.999999696  
## Cycle 134 0.000000039836231 0.00000021892573 0.999999741  
## Cycle 135 0.000000033356654 0.00000018678171 0.999999780  
## Cycle 136 0.000000027916793 0.00000015928814 0.999999813  
## Cycle 137 0.000000023352233 0.00000013578316 0.999999841  
## Cycle 138 0.000000019524149 0.00000011569739 0.999999865  
## Cycle 139 0.000000016315397 0.00000009854130 0.999999885  
## Cycle 140 0.000000013627179 0.00000008389417 0.999999902  
## Cycle 141 0.000000011376221 0.00000007139469 0.999999917  
## Cycle 142 0.000000009492373 0.00000006073268 0.999999930  
## Cycle 143 0.000000007916573 0.00000005164202 0.999999940

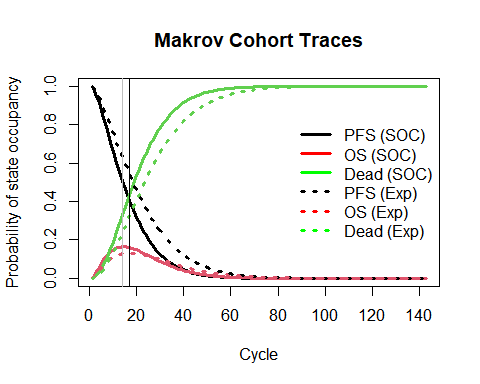
#m\_M\_Exp  
m\_M\_Exp

## PFS OS Dead  
## Cycle 1 1.000000000000 0.000000000000 0.000000000  
## Cycle 2 0.986847718283 0.010089265902 0.003063016  
## Cycle 3 0.966063070135 0.024463980960 0.009472949  
## Cycle 4 0.941303760268 0.039547468020 0.019148772  
## Cycle 5 0.913886497087 0.054185732359 0.031927771  
## Cycle 6 0.884610488280 0.067825094814 0.047564417  
## Cycle 7 0.854041044777 0.080189149801 0.065769805  
## Cycle 8 0.822609378424 0.091153683024 0.086236939  
## Cycle 9 0.790658666171 0.100684767401 0.108656566  
## Cycle 10 0.758468967539 0.108803902523 0.132727130  
## Cycle 11 0.726272251599 0.115566815013 0.158160933  
## Cycle 12 0.694262227050 0.121049930061 0.184687843  
## Cycle 13 0.662601198235 0.125341471696 0.212057330  
## Cycle 14 0.631425102195 0.128535500967 0.240039397  
## Cycle 15 0.600847371267 0.130727887989 0.268424741  
## Cycle 16 0.570962001633 0.132013591107 0.297024407  
## Cycle 17 0.541846062993 0.132484836743 0.325669100  
## Cycle 18 0.513561800383 0.132229928469 0.354208271  
## Cycle 19 0.486158428449 0.131332499720 0.382509072  
## Cycle 20 0.459673686730 0.129871081025 0.410455232  
## Cycle 21 0.434135204079 0.127918890581 0.437945905  
## Cycle 22 0.409561706836 0.125543783070 0.464894510  
## Cycle 23 0.385964096244 0.122808309884 0.491227594  
## Cycle 24 0.363346414306 0.119769856752 0.516883729  
## Cycle 25 0.341706712885 0.116480834059 0.541812453  
## Cycle 26 0.321037837676 0.112988901769 0.565973261  
## Cycle 27 0.301328136454 0.109337215734 0.589334648  
## Cycle 28 0.282562099277 0.105564685716 0.611873215  
## Cycle 29 0.264720937151 0.101706238078 0.633572825  
## Cycle 30 0.247783104657 0.097793077992 0.654423817  
## Cycle 31 0.231724771381 0.093852947504 0.674422281  
## Cycle 32 0.216520246391 0.089910376800 0.693569377  
## Cycle 33 0.202142359582 0.085986926849 0.711870714  
## Cycle 34 0.188562803332 0.082101422180 0.729335774  
## Cycle 35 0.175752437635 0.078270172986 0.745977389  
## Cycle 36 0.163681561616 0.074507186091 0.761811252  
## Cycle 37 0.152320154109 0.070824364553 0.776855481  
## Cycle 38 0.141638085813 0.067231695865 0.791130218  
## Cycle 39 0.131605305356 0.063737428857 0.804657266  
## Cycle 40 0.122192001436 0.060348239498 0.817459759  
## Cycle 41 0.113368743103 0.057069385877 0.829561871  
## Cycle 42 0.105106600066 0.053904852699 0.840988547  
## Cycle 43 0.097377244826 0.050857485657 0.851765270  
## Cycle 44 0.090153038297 0.047929116094 0.861917846  
## Cycle 45 0.083407100491 0.045120676364 0.871472223  
## Cycle 46 0.077113367694 0.042432306323 0.880454326  
## Cycle 47 0.071246637522 0.039863451385 0.888889911  
## Cycle 48 0.065782603091 0.037412952579 0.896804444  
## Cycle 49 0.060697877486 0.035079129034 0.904222993  
## Cycle 50 0.055970009608 0.032859853333 0.911170137  
## Cycle 51 0.051577492396 0.030752620139 0.917669887  
## Cycle 52 0.047499764354 0.028754608512 0.923745627  
## Cycle 53 0.043717205228 0.026862738322 0.929420056  
## Cycle 54 0.040211126599 0.025073721132 0.934715152  
## Cycle 55 0.036963758130 0.023384105934 0.939652136  
## Cycle 56 0.033958230082 0.021790320095 0.944251450  
## Cycle 57 0.031178552715 0.020288705858 0.948532741  
## Cycle 58 0.028609593088 0.018875552735 0.952514854  
## Cycle 59 0.026237049759 0.017547126098 0.956215824  
## Cycle 60 0.024047425790 0.016299692273 0.959652882  
## Cycle 61 0.022028000478 0.015129540427 0.962842459  
## Cycle 62 0.020166800128 0.014033001512 0.965800198  
## Cycle 63 0.018452568193 0.013006464515 0.968540967  
## Cycle 64 0.016874735051 0.012046390280 0.971078875  
## Cycle 65 0.015423387637 0.011149323088 0.973427289  
## Cycle 66 0.014089239174 0.010311900243 0.975598861  
## Cycle 67 0.012863599142 0.009530859842 0.977605541  
## Cycle 68 0.011738343668 0.008803046914 0.979458609  
## Cycle 69 0.010705886460 0.008125418110 0.981168695  
## Cycle 70 0.009759150388 0.007495045086 0.982745805  
## Cycle 71 0.008891539817 0.006909116748 0.984199343  
## Cycle 72 0.008096913745 0.006364940468 0.985538146  
## Cycle 73 0.007369559836 0.005859942423 0.986770498  
## Cycle 74 0.006704169364 0.005391667147 0.987904163  
## Cycle 75 0.006095813111 0.004957776420 0.988946410  
## Cycle 76 0.005539918254 0.004556047574 0.989904034  
## Cycle 77 0.005032246236 0.004184371318 0.990783382  
## Cycle 78 0.004568871632 0.003840749141 0.991590379  
## Cycle 79 0.004146162009 0.003523290389 0.992330548  
## Cycle 80 0.003760758779 0.003230209054 0.993009032  
## Cycle 81 0.003409559010 0.002959820351 0.993630621  
## Cycle 82 0.003089698202 0.002710537129 0.994199765  
## Cycle 83 0.002798533981 0.002480866152 0.994720600  
## Cycle 84 0.002533630708 0.002269404309 0.995196965  
## Cycle 85 0.002292744966 0.002074834772 0.995632420  
## Cycle 86 0.002073811881 0.001895923133 0.996030265  
## Cycle 87 0.001874932279 0.001731513569 0.996393554  
## Cycle 88 0.001694360613 0.001580525022 0.996725114  
## Cycle 89 0.001530493648 0.001441947448 0.997027559  
## Cycle 90 0.001381859860 0.001314838132 0.997303302  
## Cycle 91 0.001247109528 0.001198318089 0.997554572  
## Cycle 92 0.001125005466 0.001091568560 0.997783426  
## Cycle 93 0.001014414384 0.000993827620 0.997991758  
## Cycle 94 0.000914298832 0.000904386894 0.998181314  
## Cycle 95 0.000823709694 0.000822588393 0.998353702  
## Cycle 96 0.000741779215 0.000747821480 0.998510399  
## Cycle 97 0.000667714517 0.000679519950 0.998652766  
## Cycle 98 0.000600791579 0.000617159248 0.998782049  
## Cycle 99 0.000540349654 0.000560253811 0.998899397  
## Cycle 100 0.000485786103 0.000508354534 0.999005859  
## Cycle 101 0.000436551605 0.000461046369 0.999102402  
## Cycle 102 0.000392145734 0.000417946041 0.999189908  
## Cycle 103 0.000352112864 0.000378699888 0.999269187  
## Cycle 104 0.000316038400 0.000342981821 0.999340980  
## Cycle 105 0.000283545290 0.000310491397 0.999405963  
## Cycle 106 0.000254290820 0.000280952004 0.999464757  
## Cycle 107 0.000227963654 0.000254109153 0.999517927  
## Cycle 108 0.000204281120 0.000229728873 0.999565990  
## Cycle 109 0.000182986704 0.000207596207 0.999609417  
## Cycle 110 0.000163847757 0.000187513800 0.999648638  
## Cycle 111 0.000146653386 0.000169300583 0.999684046  
## Cycle 112 0.000131212515 0.000152790532 0.999715997  
## Cycle 113 0.000117352120 0.000137831524 0.999744816  
## Cycle 114 0.000104915595 0.000124284260 0.999770800  
## Cycle 115 0.000093761273 0.000112021262 0.999794217  
## Cycle 116 0.000083761063 0.000100925947 0.999815313  
## Cycle 117 0.000074799205 0.000090891759 0.999834309  
## Cycle 118 0.000066771138 0.000081821364 0.999851407  
## Cycle 119 0.000059582458 0.000073625911 0.999866792  
## Cycle 120 0.000053147979 0.000066224333 0.999880628  
## Cycle 121 0.000047390863 0.000059542714 0.999893066  
## Cycle 122 0.000042241841 0.000053513695 0.999904244  
## Cycle 123 0.000037638493 0.000048075928 0.999914286  
## Cycle 124 0.000033524598 0.000043173568 0.999923302  
## Cycle 125 0.000029849544 0.000038755812 0.999931395  
## Cycle 126 0.000026567786 0.000034776468 0.999938656  
## Cycle 127 0.000023638360 0.000031193555 0.999945168  
## Cycle 128 0.000021024438 0.000027968947 0.999951007  
## Cycle 129 0.000018692925 0.000025068031 0.999956239  
## Cycle 130 0.000016614096 0.000022459403 0.999960927  
## Cycle 131 0.000014761261 0.000020114586 0.999965124  
## Cycle 132 0.000013110470 0.000018007766 0.999968882  
## Cycle 133 0.000011640235 0.000016115562 0.999972244  
## Cycle 134 0.000010331294 0.000014416799 0.999975252  
## Cycle 135 0.000009166380 0.000012892318 0.999977941  
## Cycle 136 0.000008130023 0.000011524785 0.999980345  
## Cycle 137 0.000007208373 0.000010298531 0.999982493  
## Cycle 138 0.000006389029 0.000009199394 0.999984412  
## Cycle 139 0.000005660899 0.000008214583 0.999986125  
## Cycle 140 0.000005014058 0.000007332552 0.999987653  
## Cycle 141 0.000004439638 0.000006542879 0.999989017  
## Cycle 142 0.000003929710 0.000005836166 0.999990234  
## Cycle 143 0.000003477194 0.000005203940 0.999991319

# 06 Compute and Plot Epidemiological Outcomes

## 06.1 Cohort trace

# So, we'll plot the above Markov model for standard of care (m\_M\_SoC) to show our cohort distribution over time, i.e. the proportion of our cohort in the different health states over time.  
  
# If I wanted to do the same for Treatment A and Treatment B, I would just copy this code chunk and replace m\_M\_SoC with m\_M\_trtA and m\_M\_trtB  
  
# matplot(m\_M\_SoC, type = 'l',   
 # ylab = "Probability of state occupancy",  
 # xlab = "Cycle",  
 # main = "Cohort Trace", lwd = 3) # create a plot of the data  
# legend("right", v\_names\_states, col = c("black", "red", "green"),   
 # lty = 1:3, bty = "n") # add a legend to the graph  
  
# plot a vertical line that helps identifying at which cycle the prevalence of OS is highest  
# abline(v = which.max(m\_M\_SoC[, "OS"]), col = "gray")  
# The vertical line shows you when your sick (OS) population is the greatest that it will ever be, but it can be changed from which.max to other things (so it is finding which cycle the proportion sick is the highest and putting a vertical line there).  
  
# So, you can see in the graph everyone starts in the PFS state, but that this falls over time as people progress and leave this state, then you see OS start to peak up but then fall again as people leave this state to go into the dead state, which is an absorbing state and by the end will include everyone.  
  
# Plotting the Markov cohort traces  
matplot(m\_M\_SoC,   
 type = "l",   
 ylab = "Probability of state occupancy",  
 xlab = "Cycle",  
 main = "Makrov Cohort Traces",  
 lwd = 3,  
 lty = 1) # create a plot of the data  
matplot(m\_M\_Exp,   
 type = "l",   
 lwd = 3,  
 lty = 3,  
 add = TRUE) # add a plot of the experimental data ontop of the above plot  
legend("right",   
 legend = c(paste(v\_names\_states, "(SOC)"), paste(v\_names\_states, "(Exp)")),   
 col = rep(c("black", "red", "green"), 2),   
 lty = c(1, 1, 1, 3, 3, 3), # Line type, full (1) or dashed (3), I have entered this 12 times here because we have 6 lines under standard of care (6 full lines) and 6 lines under experimental treatment (6 dashed lines)  
 lwd = 3,  
 bty = "n")  
  
# plot a vertical line that helps identifying at which cycle the prevalence of OS is highest  
abline(v = which.max(m\_M\_SoC[, "OS"]), col = "gray")  
abline(v = which.max(m\_M\_Exp[, "OS"]), col = "black")

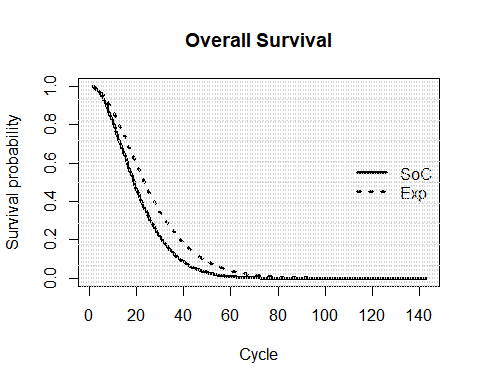


# The vertical line shows you when your sick (OS) population is the greatest that it will ever be, but it can be changed from which.max to other things (so it is finding which cycle the proportion sick is the highest and putting a vertical line there).  
  
# (It's probably not necessary for my own analysis and I can comment these two lines out if I'm not going to use it).  
  
# So, you can see in the graph everyone starts in the PFS state, but that this falls over time as people progress and leave this state, then you see OS start to peak up but then fall again as people leave this state to go into the dead state, which is an absorbing state and by the end will include everyone.  
  
  
# There is a lot going on with this plot, and that gives me two options for when I do it with the actual data:  
  
# 1. Do two separate plots for soc and exp, like in the plot I commented out at the start of this chunk.  
  
# 2. copy your soc and exp matrixes but get rid of the adverse event states in the copy in order to simplify things, i.e. reasssign all the people in the "adverse event" states to be in the PFS state, because we know that only people from PFS were allowed to be in an adverse event.  
  
  
  
  
#   
#   
#   
# matplot(m\_M\_SoC, type = 'l',   
# ylab = "Probability of state occupancy",  
# xlab = "Cycle",  
# main = "Cohort Trace", lwd = 3) # create a plot of the data  
# legend("right", v\_names\_states, col = c("black", "red", "green"),   
# lty = 1:3, bty = "n") # add a legend to the graph

## 06.2 Overall Survival (OS)

Although in the context of my analysis this would be PFS + OS because it is drawn from the DARTH model where healthy and sick make up OS, while dead means not OS (obviously).

# v\_os <- 1 - m\_M\_SoC[, "Dead"] # calculate the overall survival (OS) probability  
# v\_os <- rowSums(m\_M\_SoC[, 1:2]) # alternative way of calculating the OS probability  
  
# I could do my own version of this and chose just to look at pfs, rather than column 1 and 2 to look at anyone not dead.  
  
# i.e. v\_os <- (m\_M\_SoC[, 1])  
  
# best practice would be to rename v\_os if I am looking at something that isnt os, i.e. v\_pfs and to of course update the table legend, bearing in mind that yet again this is all for standard of care, and that if I wanted to know this for treatment a and/or treatment b I would need to replace the Markov model matrix above.  
  
  
# plot(v\_os, type = 'l',   
# ylim = c(0, 1),  
# ylab = "Survival probability",  
# xlab = "Cycle",  
# main = "Overall Survival") # create a simple plot showing the OS  
  
# add grid   
# grid(nx = n\_cycles, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"),   
# equilogs = TRUE)   
  
# Calculating and plotting overal survival (OS)  
v\_OS\_SoC <- 1 - m\_M\_SoC[, "Dead"]  
v\_OS\_Exp <- 1 - m\_M\_Exp[, "Dead"]  
  
plot(v\_OS\_SoC,   
 type = "l",  
 ylim = c(0, 1),  
 ylab = "Survival probability",  
 xlab = "Cycle",  
 main = "Overall Survival",  
 lwd = 3) # create a simple plot showing the OS  
lines(v\_OS\_Exp,  
 lty = 3,  
 lwd = 3)  
legend("right",  
 legend = c("SoC", "Exp"),  
 lty = c(1, 3),  
 lwd = 3,  
 bty = "n")  
  
# add grid - completely optional, see if it looks nicer to leave this code in or out:  
grid(nx = n\_cycle, ny = 10, col = "lightgray", lty = "dotted", lwd = par("lwd"),   
 equilogs = TRUE)



# Per 1:12 hour mark of: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \\_ DARTH\August\\_25\Live Session Recording\Live Session Recording August 25th WITH CHAT.mkv  
  
# Often you have a survival curve as input to your model [I guess from a published study], and having that survival curve you need to parameterise your model so that you match that survival curve, and that would be a process of, potentially of calibration, if you can't use the parameters directly in your model.  
  
# So, you could produce your survival curve and compare it to curves from trials, etc., to calibrate your model. So. we'll probably do this and have it in an appendix section.  
  
# For calibration purposes, you want to make sure that your model is outputting something that's comparable to the publications out there on actual data on the same type of patients.  
  
# Part of being comparable to the real world is that if there is a censoring process in actual patient data, then you could incorporate this process into the model to reflect that in your model and to ensure that your own model is comparable to the existing models which may be losing people due to censoring, etc., and which you'll then need to incorporate into your model to be comparable.   
  
  
# Another interesting thing you can do is, plot this to ask is that reasonable, does that make sense that this many people are alive after this amount of time? Is the OS what I would expect it to be?

## 06.2.1 Life Expectancy (LE)

v\_le <- sum(v\_OS\_SoC) # summing probability of OS over time (i.e. life expectancy)  
  
# Basically we are summing all the alive states over time through over all the cycles, so   
  
# v\_os <- rowSums(m\_M\_SoC[, 1:2])  
  
# Is basically the PFS and OS added together.  
  
# Also bear in mind that this is life expectancy under standard of care, and not under either of the new treatments, per: # v\_os <- rowSums(m\_M\_SoC[, 1:2]) above.  
  
v\_le

## [1] 20.70332

v\_le\_exp <- sum(v\_OS\_Exp) # summing probability of OS over time (i.e. life expectancy)  
  
v\_le\_exp

## [1] 25.94259

# So, if this gives a value of [1] 20.70332, that is [1] 20.70332 cycles, where in the context of our model, cycles are fortnights, so the life expectancy for our population of patients is 20.70332 fortnights or, in days:  
  
  
daily\_v\_le <- v\_le \* 14  
daily\_v\_le

## [1] 289.8465

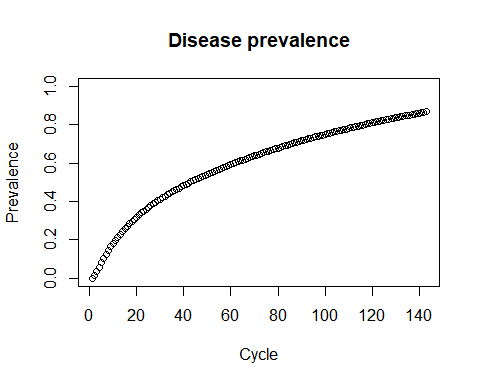
daily\_v\_le\_exp <- v\_le\_exp \* 14  
daily\_v\_le\_exp

## [1] 363.1962

# When I calculate LE I calculate it in cycles. Note that my code gives a LE of 20.7 cycles for the SoC group which is approx. 290 days or 0.8 years. Caculating the LE for the Exp group I calculated the corresponding figures as: 25.9 cycles = 363 days = 1 year (approx.). So a LE gain of about 0.2 life-years.  
  
  
  
  
  
  
  
  
  
  
# Discounted life expectancy:  
  
# If you wanted discounted life expectancy, if you were using life years and you wanted them discounted for your health economic outcomes, you could apply the discount rates - the discount factors - to the vector for overall survival [v\_os] and then take it's sum [add it up] as above to get life expectancy that is discounted.  
  
# As the 1:12 hour mark of: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Cost-Effectiveness and Decision Modeling using R Workshop \_ DARTH\August\_25\Live Session Recording\Live Session Recording August 25th WITH CHAT.mkv

## 06.3 Disease prevalence

# Disease prevalence is the proportion who are sick divided by the proportion who are alive, so it's necessary to account for the fact that some of the cohort have died, so you only calculate prevalence among people who are alive,in the diagram you can see it plateauing over time, even though the number of people in the OS (or "progressed") state have gone up and come down over time and this is because this is prevalence as a proportion of those who are alive, and there are few people who are still alive by cycle 60.  
  
# Probably looks a bit funny dividing OS by v\_os below, but it's necessary to remember that v\_os is PFS + OS, because it's anyone who is not dead.  
  
# So, I guess in our context you can think of this as progression prevalence over time:  
  
v\_prev <- m\_M\_SoC[, "OS"]/v\_OS\_SoC  
plot(v\_prev,  
 ylim = c(0, 1),  
 ylab = "Prevalence",  
 xlab = "Cycle",  
 main = "Disease prevalence")



# 07 Compute Cost-Effectiveness Outcomes

Good mathematical notation here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Applied Cost Effectiveness Modeling with R\rcea-master\rcea-master\vignettes

## 07.1 Including ADVERSE EVENTS in Mean Costs and QALYs for each strategy

# Calculate the costs and QALYs per cycle by multiplying m\_M (the Markov trace) with the cost/utility vectors for the different states  
  
# per cycle  
# calculate expected costs by multiplying cohort trace with the cost vector for the different health states  
  
# Basically, you take the cohort trace over time for each strategy [m\_M\_SoC] and multiply this by a vector of our costs for each state: [c(c\_H, c\_S, c\_D)] -> So, basically the number of people in each state at each cycle multiplied by the cost of being in that state [cost of healthy, cost of sick and cost of dead] for each strategy that we look at (standard of care, treatment A and treatment B).  
  
# Bear in mind we are doing matrix multiplication [%\*%], because what this does is for each cycle [row in the matrix], take the vector of costs, and multiply it by the distribution in that cycle [breakdown of proportions in the states in that row] and add it all together, to get the total costs for that cycle [row]. This gives us a vector of the costs accrued in this cohort of individuals ending up in these different states for each cycle on a per person basis, because it's a cohort distribution (it always sums to 1).  
  
# So, in cycle 1 everyone is in the OS state, so they only incur the OS cost, as more and more time passes and more and more people get sick, the costs increases due to more people being in the PFS state, but over time this falls again as more and more people go into the dead state, which has no costs as we don't treat corpses.  
  
# v\_tc\_SoC <- m\_M\_SoC %\*% c(c\_H, c\_S, c\_D)   
# v\_tc\_trtA <- m\_M\_trtA %\*% c(c\_H + c\_trtA, c\_S, c\_D)   
# v\_tc\_trtB <- m\_M\_trtB %\*% c(c\_H + c\_trtB, c\_S, c\_D)   
  
# 1. Probability of adverse events 1,2 3 in the PFS state under standard of care and exp care:  
  
# AE1:Leukopenia = 0.040 AE2:Diarrhea = 0.310 AE3:Vomiting = 0.31  
  
p\_FA1\_STD <- 0.040 # probability of adverse event 1 when progression-free under SOC  
p\_FA2\_STD <- 0.310 # probability of adverse event 2 when progression-free under SOC  
p\_FA3\_STD <- 0.310 # probability of adverse event 3 when progression-free under SOC  
  
p\_FA1\_EXPR <- 0.070 # probability of adverse event 1 when progression-free under EXPR  
p\_FA2\_EXPR <- 0.110 # probability of adverse event 2 when progression-free under EXPR  
p\_FA3\_EXPR <- 0.070 # probability of adverse event 3 when progression-free under EXPR  
  
# 2. Cost of treating the AE conditional on it occurring  
c\_AE1 <- 2835.89  
c\_AE2 <- 1458.80  
c\_AE3 <- 409.03   
#3. Disutiltiy of AE (negative qol impact x duration of event)  
  
  
AE1\_DisUtil <-0.45  
AE2\_DisUtil <-0.19  
AE3\_DisUtil <-0.36  
  
daily\_utility <- u\_F/14  
AE1\_discounted\_daily\_utility <- daily\_utility \* (1-AE1\_DisUtil)  
AE2\_discounted\_daily\_utility <- daily\_utility \* (1-AE2\_DisUtil)  
AE3\_discounted\_daily\_utility <- daily\_utility \* (1-AE3\_DisUtil)  
  
  
u\_AE1 <- (AE1\_discounted\_daily\_utility\*7) + (daily\_utility\*7)  
u\_AE2 <- (AE2\_discounted\_daily\_utility\*7) + (daily\_utility\*7)  
u\_AE3 <- (AE3\_discounted\_daily\_utility\*7) + (daily\_utility\*7)  
  
  
# I then adjust my state costs and utilities:  
  
# uState<-uState-pAE1\*duAE1  
  
  
c\_F\_SoC<-c\_F\_SoC +p\_FA1\_STD\*c\_AE1 +p\_FA2\_STD\*c\_AE2 +p\_FA3\_STD\*c\_AE3  
c\_F\_Exp<-c\_F\_Exp +p\_FA1\_EXPR\*c\_AE1 +p\_FA2\_EXPR\*c\_AE2 +p\_FA3\_EXPR\*c\_AE3  
  
  
u\_F\_SoC<-u\_F  
u\_F\_Exp<-u\_F  
  
  
u\_F\_SoC<-u\_F-p\_FA1\_STD\*u\_AE1 -p\_FA2\_STD\*u\_AE2 -p\_FA3\_STD\*u\_AE3  
u\_F\_Exp<-u\_F-p\_FA1\_EXPR\*u\_AE1 -p\_FA2\_EXPR\*u\_AE2 -p\_FA3\_EXPR\*u\_AE3  
  
  
v\_tc\_SoC <- m\_M\_SoC %\*% c(c\_F\_SoC, c\_P, c\_D)  
v\_tc\_Exp <- m\_M\_Exp %\*% c(c\_F\_Exp, c\_P, c\_D)  
  
v\_tc\_SoC

## [,1]  
## Cycle 1 1365.27290000000  
## Cycle 2 1348.88093020235  
## Cycle 3 1321.61012829862  
## Cycle 4 1287.43169885667  
## Cycle 5 1247.99913474360  
## Cycle 6 1204.50164263568  
## Cycle 7 1157.91210127763  
## Cycle 8 1109.06292817166  
## Cycle 9 1058.67737275735  
## Cycle 10 1007.38572039179  
## Cycle 11 955.73561217531  
## Cycle 12 904.19985082165  
## Cycle 13 853.18304028956  
## Cycle 14 803.02761919057  
## Cycle 15 754.01951821648  
## Cycle 16 706.39352861953  
## Cycle 17 660.33840667114  
## Cycle 18 616.00171366074  
## Cycle 19 573.49438257793  
## Cycle 20 532.89500192931  
## Cycle 21 494.25380976016  
## Cycle 22 457.59639474248  
## Cycle 23 422.92710512672  
## Cycle 24 390.23216996383  
## Cycle 25 359.48254009356  
## Cycle 26 330.63645891256  
## Cycle 27 303.64177489800  
## Cycle 28 278.43800931784  
## Cycle 29 254.95819356593  
## Cycle 30 233.13049118337  
## Cycle 31 212.87961992480  
## Cycle 32 194.12808925558  
## Cycle 33 176.79726847182  
## Cycle 34 160.80830026389  
## Cycle 35 146.08287403267  
## Cycle 36 132.54387265004  
## Cycle 37 120.11590565855  
## Cycle 38 108.72574115434  
## Cycle 39 98.30264781254  
## Cycle 40 88.77865771241  
## Cycle 41 80.08875981487  
## Cycle 42 72.17103314946  
## Cycle 43 64.96672799021  
## Cycle 44 58.42030254838  
## Cycle 45 52.47942198961  
## Cycle 46 47.09492589888  
## Cycle 47 42.22076967052  
## Cycle 48 37.81394469589  
## Cycle 49 33.83438165774  
## Cycle 50 30.24484071878  
## Cycle 51 27.01079191241  
## Cycle 52 24.10028860405  
## Cycle 53 21.48383649197  
## Cycle 54 19.13426025444  
## Cycle 55 17.02656962400  
## Cycle 56 15.13782637756  
## Cycle 57 13.44701347083  
## Cycle 58 11.93490731504  
## Cycle 59 10.58395399050  
## Cycle 60 9.37815001417  
## Cycle 61 8.30292812316  
## Cycle 62 7.34504840296  
## Cycle 63 6.49249497426  
## Cycle 64 5.73437835510  
## Cycle 65 5.06084353308  
## Cycle 66 4.46298371421  
## Cycle 67 3.93275965902  
## Cycle 68 3.46292447112  
## Cycle 69 3.04695366779  
## Cycle 70 2.67898033450  
## Cycle 71 2.35373514446  
## Cycle 72 2.06649101033  
## Cycle 73 1.81301212550  
## Cycle 74 1.58950714783  
## Cycle 75 1.39258627739  
## Cycle 76 1.21922198168  
## Cycle 77 1.06671312598  
## Cycle 78 0.93265227305  
## Cycle 79 0.81489592388  
## Cycle 80 0.71153748064  
## Cycle 81 0.62088272242  
## Cycle 82 0.54142759525  
## Cycle 83 0.47183812843  
## Cycle 84 0.41093230024  
## Cycle 85 0.35766368731  
## Cycle 86 0.31110674248  
## Cycle 87 0.27044355688  
## Cycle 88 0.23495197222  
## Cycle 89 0.20399491909  
## Cycle 90 0.17701086691  
## Cycle 91 0.15350527975  
## Cycle 92 0.13304298143  
## Cycle 93 0.11524134075  
## Cycle 94 0.09976419587  
## Cycle 95 0.08631644350  
## Cycle 96 0.07463922546  
## Cycle 97 0.06450565121  
## Cycle 98 0.05571700058  
## Cycle 99 0.04809935624  
## Cycle 100 0.04150062027  
## Cycle 101 0.03578787364  
## Cycle 102 0.03084504137  
## Cycle 103 0.02657083002  
## Cycle 104 0.02287690734  
## Cycle 105 0.01968629720  
## Cycle 106 0.01693196561  
## Cycle 107 0.01455557607  
## Cycle 108 0.01250639508  
## Cycle 109 0.01074033039  
## Cycle 110 0.00921908669  
## Cycle 111 0.00790942494  
## Cycle 112 0.00678251317  
## Cycle 113 0.00581335802  
## Cycle 114 0.00498030711  
## Cycle 115 0.00426461411  
## Cycle 116 0.00365005854  
## Cycle 117 0.00312261396  
## Cycle 118 0.00267015842  
## Cycle 119 0.00228222207  
## Cycle 120 0.00194976728  
## Cycle 121 0.00166499721  
## Cycle 122 0.00142118933  
## Cycle 123 0.00121255066  
## Cycle 124 0.00103409204  
## Cycle 125 0.00088151894  
## Cycle 126 0.00075113682  
## Cycle 127 0.00063976902  
## Cycle 128 0.00054468558  
## Cycle 129 0.00046354175  
## Cycle 130 0.00039432461  
## Cycle 131 0.00033530709  
## Cycle 132 0.00028500807  
## Cycle 133 0.00024215803  
## Cycle 134 0.00020566939  
## Cycle 135 0.00017461083  
## Cycle 136 0.00014818533  
## Cycle 137 0.00012571105  
## Cycle 138 0.00010660500  
## Cycle 139 0.00009036897  
## Cycle 140 0.00007657737  
## Cycle 141 0.00006486681  
## Cycle 142 0.00005492718  
## Cycle 143 0.00004649395

v\_tc\_Exp

## [,1]  
## Cycle 1 3640.80240000  
## Cycle 2 3599.88942568  
## Cycle 3 3534.14984442  
## Cycle 4 3454.42908086  
## Cycle 5 3364.72357670  
## Cycle 6 3267.56048581  
## Cycle 7 3164.80699182  
## Cycle 8 3057.94721727  
## Cycle 9 2948.20715735  
## Cycle 10 2836.62131006  
## Cycle 11 2724.07273719  
## Cycle 12 2611.31950515  
## Cycle 13 2499.01349655  
## Cycle 14 2387.71462937  
## Cycle 15 2277.90213650  
## Cycle 16 2169.98385758  
## Cycle 17 2064.30411846  
## Cycle 18 1961.15056055  
## Cycle 19 1860.76015703  
## Cycle 20 1763.32457627  
## Cycle 21 1668.99500470  
## Cycle 22 1577.88651017  
## Cycle 23 1490.08200621  
## Cycle 24 1405.63586365  
## Cycle 25 1324.57720632  
## Cycle 26 1246.91292080  
## Cycle 27 1172.63040520  
## Cycle 28 1101.70007832  
## Cycle 29 1034.07766795  
## Cycle 30 969.70629487  
## Cycle 31 908.51836757  
## Cycle 32 850.43730129  
## Cycle 33 795.37907410  
## Cycle 34 743.25363168  
## Cycle 35 693.96615168  
## Cycle 36 647.41817810  
## Cycle 37 603.50863504  
## Cycle 38 562.13472924  
## Cycle 39 523.19274968  
## Cycle 40 486.57877255  
## Cycle 41 452.18927900  
## Cycle 42 419.92169309  
## Cycle 43 389.67484641  
## Cycle 44 361.34937600  
## Cycle 45 334.84806143  
## Cycle 46 310.07610649  
## Cycle 47 286.94137106  
## Cycle 48 265.35455770  
## Cycle 49 245.22935777  
## Cycle 50 226.48256116  
## Cycle 51 209.03413367  
## Cycle 52 192.80726563  
## Cycle 53 177.72839515  
## Cycle 54 163.72720901  
## Cycle 55 150.73662420  
## Cycle 56 138.69275257  
## Cycle 57 127.53485107  
## Cycle 58 117.20525963  
## Cycle 59 107.64932881  
## Cycle 60 98.81533889  
## Cycle 61 90.65441203  
## Cycle 62 83.12041901  
## Cycle 63 76.16988167  
## Cycle 64 69.76187248  
## Cycle 65 63.85791197  
## Cycle 66 58.42186511  
## Cycle 67 53.41983740  
## Cycle 68 48.82007128  
## Cycle 69 44.59284354  
## Cycle 70 40.71036421  
## Cycle 71 37.14667736  
## Cycle 72 33.87756416  
## Cycle 73 30.88044855  
## Cycle 74 28.13430574  
## Cycle 75 25.61957366  
## Cycle 76 23.31806767  
## Cycle 77 21.21289844  
## Cycle 78 19.28839327  
## Cycle 79 17.53002072  
## Cycle 80 15.92431865  
## Cycle 81 14.45882569  
## Cycle 82 13.12201600  
## Cycle 83 11.90323736  
## Cycle 84 10.79265253  
## Cycle 85 9.78118370  
## Cycle 86 8.86046008  
## Cycle 87 8.02276845  
## Cycle 88 7.26100659  
## Cycle 89 6.56863947  
## Cycle 90 5.93965814  
## Cycle 91 5.36854113  
## Cycle 92 4.85021831  
## Cycle 93 4.38003709  
## Cycle 94 3.95373081  
## Cycle 95 3.56738926  
## Cycle 96 3.21743115  
## Cycle 97 2.90057849  
## Cycle 98 2.61383280  
## Cycle 99 2.35445290  
## Cycle 100 2.11993436  
## Cycle 101 1.90799039  
## Cycle 102 1.71653420  
## Cycle 103 1.54366256  
## Cycle 104 1.38764066  
## Cycle 105 1.24688814  
## Cycle 106 1.11996608  
## Cycle 107 1.00556513  
## Cycle 108 0.90249444  
## Cycle 109 0.80967156  
## Cycle 110 0.72611309  
## Cycle 111 0.65092609  
## Cycle 112 0.58330015  
## Cycle 113 0.52250022  
## Cycle 114 0.46785986  
## Cycle 115 0.41877520  
## Cycle 116 0.37469933  
## Cycle 117 0.33513715  
## Cycle 118 0.29964072  
## Cycle 119 0.26780493  
## Cycle 120 0.23926363  
## Cycle 121 0.21368597  
## Cycle 122 0.19077323  
## Cycle 123 0.17025574  
## Cycle 124 0.15189024  
## Cycle 125 0.13545733  
## Cycle 126 0.12075929  
## Cycle 127 0.10761797  
## Cycle 128 0.09587293  
## Cycle 129 0.08537976  
## Cycle 130 0.07600854  
## Cycle 131 0.06764242  
## Cycle 132 0.06017636  
## Cycle 133 0.05351597  
## Cycle 134 0.04757650  
## Cycle 135 0.04228183  
## Cycle 136 0.03756366  
## Cycle 137 0.03336075  
## Cycle 138 0.02961816  
## Cycle 139 0.02628665  
## Cycle 140 0.02332214  
## Cycle 141 0.02068510  
## Cycle 142 0.01834021  
## Cycle 143 0.01625580

# calculate expected QALYs by multiplying cohort trace with the utilities for the different health states   
  
# The three vectors of utilities are basically built in the exact same way as the three vectors of costs above:  
  
# v\_tu\_SoC <- m\_M\_SoC %\*% c(u\_H, u\_S, u\_D)   
# v\_tu\_trtA <- m\_M\_trtA %\*% c(u\_H, u\_S, u\_D)   
# v\_tu\_trtB <- m\_M\_trtB %\*% c(u\_H, u\_S, u\_D)   
  
# The file I am mirroring has the following qoute:  
  
# "  
  
# - note that to obtain QALYs, the utility needs to be mutiplied by the cycle length as well  
  
  
# v\_tu\_SoC <- m\_M\_SoC %\*% c(u\_F, u\_P, u\_D) \* t\_cycle  
# v\_tu\_Exp <- m\_M\_Exp %\*% c(u\_F, u\_P, u\_D) \* t\_cycle  
  
# To get the QALY's we not only need to multiply the state occupancy with the utilities, but also with the duration of the time cycle, because QALY's are 2-dimensional in that they combine the duration of time and the health utility.  
  
# "  
  
# Maybe that's because the utility originally came from a year in the disease state, and we want it to reflect the proportion of a year that is our cycle length?  
  
# So in the example t\_cycle <- 1/4 # cycle length of 3 months (in years) - so maybe their utility originally came from a years utility and they wanted to decrease it to 3 months, aka the cycle lengths, utility.  
  
# If that was the case it would be:  
  
# A year of utility in this state is 0.75, so 3 months should be a quarter of this, should be 0.25 of this. So, t\_cycle (0.25) \* u\_F (0.75) = 0.1875 i.e. your utility for 3 months if u\_F is your utility for 12 months.  
  
# v\_tu\_SoC <- m\_M\_SoC %\*% c(u\_F, u\_P, u\_D) \* t\_cycle  
  
# I did the maths on his approach, and without \*t\_cycle the first cycle of each utility value is 0.8, but after \*t\_cycle it is 0.2, i.e. a quarter of what it was before. Which is why I think again that he is just making the yearly utility lower to match the 3 monthly cycles, i.e. quartering a yearly utility. A quarter of utility for a quarter of a year.  
  
# So, I think he's just trying to generate the QALYs per cycle in both states.  
  
# And slide 25 of C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Health Economic Modeling in R A Hands-on Introduction\Health-Eco\Markov models kind of proves that.  
  
# In it they say: cycles are 6 months.  
  
# Their R code in:   
  
# C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Health Economic Modeling in R A Hands-on Introduction\Health-Eco\Markov models\markov\_smoking\_deterministic.R  
  
# Says:  
  
# Now define the QALYS associated with the states per cycle  
  
# QALY associated with 1 - year in the smoking state is Normal(mean = 0.95, SD = 0.01)  
# Divide by 2 as cycle length is 6 months  
# state\_qalys["Smoking"] <- 0.95 / 2 [I know they divide by 2 but we could have also multiplied by a half, i.e. \*0.05].  
  
# QALY associated with 1 - year in the not smoking state is 1 (no uncertainty)  
# So all PSA samples have the same value  
# Again divide by 2 as cycle length is 6 months  
# state\_qalys["Not smoking"] <- 1.0 / 2  
  
# I like their approach, they define utility at the start, then they can do the following:  
  
 # Now use the cohort vectors to calculate the   
 # total QALYs for each cycle  
 # cycle\_qalys[i\_treatment, ] <- cohort\_vectors[i\_treatment, , ] %\*% state\_qalys[]  
  
# i.e., take the cohort (or Markov trace) for each of the treatment options and multiply it by the state qalys.  
  
# So, I could take where he says: "QALY's are 2-dimensional in that they combine the duration of time and the health utility." as saying "we need to give the patient a utility for this health state that matches how long they were in this health state, i.e., if the utility of one year in this state is x and we know the patient was in this health state for 2 weeks, then we need to give them 2 weeks of x as their utility."  
  
  
# Andrew calculated QALYs as:   
#  
# QALYs.SP0 <- trace.SP0%\*%state.utilities  
# QALYs.SP0  
#   
# undisc.QALYs.SP0 <- colSums(QALYs.SP0)  
# undisc.QALYs.SP0  
  
# However, he described the values that he started off with as:  
  
# The quality of life utilities of a year spent in each of the model health states has been  
# estimated to be 0.85, 0.3 and 0.75 for the Successful primary, Revision, and Successful revision health states respectively.  
  
# and said the cycle length is one year, so maybe that's why he doesnt need to multiply by the time passed, because his utilities are for a year spent in this state, whereas when we start our utilies may not necessarily match.  
  
# So, what I think this means is that provided utilities are for the period of the cycle we can do the below, and if they are not for the period of the cycle then we can just convert them like in the smoking file above and then apply them as though they are for the period of the cycle:  
  
  
  
  
v\_tu\_SoC <- m\_M\_SoC %\*% c(u\_F\_SoC, u\_P, u\_D)  
v\_tu\_Exp <- m\_M\_Exp %\*% c(u\_F\_Exp, u\_P, u\_D)  
  
v\_tu\_SoC

## [,1]  
## Cycle 1 0.36911250000000  
## Cycle 2 0.37152532212412  
## Cycle 3 0.37380813764819  
## Cycle 4 0.37455093072850  
## Cycle 5 0.37339161497378  
## Cycle 6 0.37026814369279  
## Cycle 7 0.36526071092749  
## Cycle 8 0.35852361697456  
## Cycle 9 0.35024893061083  
## Cycle 10 0.34064506421544  
## Cycle 11 0.32992349859670  
## Cycle 12 0.31829039987215  
## Cycle 13 0.30594135638684  
## Cycle 14 0.29305818609772  
## Cycle 15 0.27980715316435  
## Cycle 16 0.26633815762404  
## Cycle 17 0.25278460045629  
## Cycle 18 0.23926371549519  
## Cycle 19 0.22587721924294  
## Cycle 20 0.21271217069509  
## Cycle 21 0.19984196228293  
## Cycle 22 0.18732738393778  
## Cycle 23 0.17521771760344  
## Cycle 24 0.16355183091310  
## Cycle 25 0.15235924731161  
## Cycle 26 0.14166117640177  
## Cycle 27 0.13147149325689  
## Cycle 28 0.12179765924958  
## Cycle 29 0.11264157987441  
## Cycle 30 0.10400039729087  
## Cycle 31 0.09586721703468  
## Cycle 32 0.08823176965302  
## Cycle 33 0.08108100900053  
## Cycle 34 0.07439964965452  
## Cycle 35 0.06817064642353  
## Cycle 36 0.06237561927373  
## Cycle 37 0.05699522721720  
## Cycle 38 0.05200949482030  
## Cycle 39 0.04739809502239  
## Cycle 40 0.04314059192126  
## Cycle 41 0.03921664709779  
## Cycle 42 0.03560619292945  
## Cycle 43 0.03228957619082  
## Cycle 44 0.02924767506733  
## Cycle 45 0.02646199252255  
## Cycle 46 0.02391472876497  
## Cycle 47 0.02158883536257  
## Cycle 48 0.01946805335505  
## Cycle 49 0.01753693751834  
## Cycle 50 0.01578086874591  
## Cycle 51 0.01418605632783  
## Cycle 52 0.01273953173374  
## Cycle 53 0.01142913533979  
## Cycle 54 0.01024349738398  
## Cycle 55 0.00917201428828  
## Cycle 56 0.00820482135109  
## Cycle 57 0.00733276268861  
## Cycle 58 0.00654735918923  
## Cycle 59 0.00584077514083  
## Cycle 60 0.00520578409568  
## Cycle 61 0.00463573445254  
## Cycle 62 0.00412451515830  
## Cycle 63 0.00366652186310  
## Cycle 64 0.00325662380185  
## Cycle 65 0.00289013162134  
## Cycle 66 0.00256276632456  
## Cycle 67 0.00227062946313  
## Cycle 68 0.00201017467255  
## Cycle 69 0.00177818061448  
## Cycle 70 0.00157172536415  
## Cycle 71 0.00138816225864  
## Cycle 72 0.00122509720335  
## Cycle 73 0.00108036741883  
## Cycle 74 0.00095202159732  
## Cycle 75 0.00083830142887  
## Cycle 76 0.00073762444870  
## Cycle 77 0.00064856815200  
## Cycle 78 0.00056985531771  
## Cycle 79 0.00050034048057  
## Cycle 80 0.00043899748831  
## Cycle 81 0.00038490808110  
## Cycle 82 0.00033725142972  
## Cycle 83 0.00029529457042  
## Cycle 84 0.00025838367544  
## Cycle 85 0.00022593610027  
## Cycle 86 0.00019743315066  
## Cycle 87 0.00017241351515  
## Cycle 88 0.00015046731108  
## Cycle 89 0.00013123069522  
## Cycle 90 0.00011438099249  
## Cycle 91 0.00009963229945  
## Cycle 92 0.00008673152166  
## Cycle 93 0.00007545480705  
## Cycle 94 0.00006560433986  
## Cycle 95 0.00005700546248  
## Cycle 96 0.00004950409472  
## Cycle 97 0.00004296442283  
## Cycle 98 0.00003726683232  
## Cycle 99 0.00003230606111  
## Cycle 100 0.00002798955141  
## Cycle 101 0.00002423598055  
## Cycle 102 0.00002097395270  
## Cycle 103 0.00001814083521  
## Cycle 104 0.00001568172463  
## Cycle 105 0.00001354852878  
## Cycle 106 0.00001169915288  
## Cycle 107 0.00001009677841  
## Cycle 108 0.00000870922497  
## Cycle 109 0.00000750838596  
## Cycle 110 0.00000646973009  
## Cycle 111 0.00000557186145  
## Cycle 112 0.00000479613149  
## Cycle 113 0.00000412629731  
## Cycle 114 0.00000354822075  
## Cycle 115 0.00000304960382  
## Cycle 116 0.00000261975617  
## Cycle 117 0.00000224939093  
## Cycle 118 0.00000193044562  
## Cycle 119 0.00000165592514  
## Cycle 120 0.00000141976425  
## Cycle 121 0.00000121670720  
## Cycle 122 0.00000104220250  
## Cycle 123 0.00000089231091  
## Cycle 124 0.00000076362514  
## Cycle 125 0.00000065319972  
## Cycle 126 0.00000055848994  
## Cycle 127 0.00000047729857  
## Cycle 128 0.00000040772950  
## Cycle 129 0.00000034814740  
## Cycle 130 0.00000029714263  
## Cycle 131 0.00000025350082  
## Cycle 132 0.00000021617631  
## Cycle 133 0.00000018426928  
## Cycle 134 0.00000015700578  
## Cycle 135 0.00000013372047  
## Cycle 136 0.00000011384173  
## Cycle 137 0.00000009687865  
## Cycle 138 0.00000008240991  
## Cycle 139 0.00000007007406  
## Cycle 140 0.00000005956117  
## Cycle 141 0.00000005060565  
## Cycle 142 0.00000004298000  
## Cycle 143 0.00000003648942

v\_tu\_Exp

## [,1]  
## Cycle 1 0.670480000000  
## Cycle 2 0.668219680991  
## Cycle 3 0.663627554888  
## Cycle 4 0.656831199398  
## Cycle 5 0.647963344600  
## Cycle 6 0.637199951811  
## Cycle 7 0.624740387072  
## Cycle 8 0.610793030011  
## Cycle 9 0.595565921305  
## Cycle 10 0.579260809996  
## Cycle 11 0.562069449011  
## Cycle 12 0.544171392533  
## Cycle 13 0.525732807995  
## Cycle 14 0.506905978149  
## Cycle 15 0.487829272680  
## Cycle 16 0.468627437074  
## Cycle 17 0.449412092199  
## Cycle 18 0.430282369426  
## Cycle 19 0.411325627924  
## Cycle 20 0.392618216145  
## Cycle 21 0.374226250508  
## Cycle 22 0.356206392195  
## Cycle 23 0.338606608674  
## Cycle 24 0.321466910753  
## Cycle 25 0.304820058994  
## Cycle 26 0.288692235555  
## Cycle 27 0.273103679156  
## Cycle 28 0.258069282039  
## Cycle 29 0.243599148691  
## Cycle 30 0.229699116705  
## Cycle 31 0.216371240593  
## Cycle 32 0.203614239720  
## Cycle 33 0.191423911704  
## Cycle 34 0.179793512795  
## Cycle 35 0.168714106826  
## Cycle 36 0.158174884392  
## Cycle 37 0.148163453887  
## Cycle 38 0.138666106089  
## Cycle 39 0.129668053892  
## Cycle 40 0.121153648796  
## Cycle 41 0.113106575696  
## Cycle 42 0.105510027467  
## Cycle 43 0.098346860788  
## Cycle 44 0.091599734579  
## Cycle 45 0.085251232374  
## Cycle 46 0.079283969881  
## Cycle 47 0.073680688926  
## Cycle 48 0.068424338896  
## Cycle 49 0.063498146769  
## Cycle 50 0.058885676708  
## Cycle 51 0.054570880192  
## Cycle 52 0.050538137537  
## Cycle 53 0.046772291671  
## Cycle 54 0.043258674898  
## Cycle 55 0.039983129408  
## Cycle 56 0.036932022167  
## Cycle 57 0.034092254831  
## Cycle 58 0.031451269251  
## Cycle 59 0.028997049086  
## Cycle 60 0.026718118021  
## Cycle 61 0.024603535038  
## Cycle 62 0.022642887132  
## Cycle 63 0.020826279857  
## Cycle 64 0.019144326039  
## Cycle 65 0.017588132950  
## Cycle 66 0.016149288240  
## Cycle 67 0.014819844850  
## Cycle 68 0.013592305156  
## Cycle 69 0.012459604525  
## Cycle 70 0.011415094458  
## Cycle 71 0.010452525502  
## Cycle 72 0.009566030032  
## Cycle 73 0.008750105054  
## Cycle 74 0.007999595120  
## Cycle 75 0.007309675447  
## Cycle 76 0.006675835314  
## Cycle 77 0.006093861813  
## Cycle 78 0.005559823993  
## Cycle 79 0.005070057457  
## Cycle 80 0.004621149431  
## Cycle 81 0.004209924354  
## Cycle 82 0.003833429984  
## Cycle 83 0.003488924062  
## Cycle 84 0.003173861518  
## Cycle 85 0.002885882246  
## Cycle 86 0.002622799427  
## Cycle 87 0.002382588414  
## Cycle 88 0.002163376168  
## Cycle 89 0.001963431222  
## Cycle 90 0.001781154185  
## Cycle 91 0.001615068754  
## Cycle 92 0.001463813229  
## Cycle 93 0.001326132509  
## Cycle 94 0.001200870561  
## Cycle 95 0.001086963331  
## Cycle 96 0.000983432090  
## Cycle 97 0.000889377197  
## Cycle 98 0.000803972249  
## Cycle 99 0.000726458613  
## Cycle 100 0.000656140313  
## Cycle 101 0.000592379260  
## Cycle 102 0.000534590798  
## Cycle 103 0.000482239560  
## Cycle 104 0.000434835610  
## Cycle 105 0.000391930854  
## Cycle 106 0.000353115712  
## Cycle 107 0.000318016020  
## Cycle 108 0.000286290173  
## Cycle 109 0.000257626460  
## Cycle 110 0.000231740615  
## Cycle 111 0.000208373541  
## Cycle 112 0.000187289213  
## Cycle 113 0.000168272740  
## Cycle 114 0.000151128577  
## Cycle 115 0.000135678879  
## Cycle 116 0.000121761983  
## Cycle 117 0.000109231014  
## Cycle 118 0.000097952599  
## Cycle 119 0.000087805688  
## Cycle 120 0.000078680473  
## Cycle 121 0.000070477390  
## Cycle 122 0.000063106212  
## Cycle 123 0.000056485210  
## Cycle 124 0.000050540392  
## Cycle 125 0.000045204800  
## Cycle 126 0.000040417873  
## Cycle 127 0.000036124858  
## Cycle 128 0.000032276280  
## Cycle 129 0.000028827453  
## Cycle 130 0.000025738031  
## Cycle 131 0.000022971611  
## Cycle 132 0.000020495356  
## Cycle 133 0.000018279660  
## Cycle 134 0.000016297846  
## Cycle 135 0.000014525881  
## Cycle 136 0.000012942128  
## Cycle 137 0.000011527115  
## Cycle 138 0.000010263322  
## Cycle 139 0.000009134999  
## Cycle 140 0.000008127985  
## Cycle 141 0.000007229560  
## Cycle 142 0.000006428300  
## Cycle 143 0.000005713950

# BUT, we need to remember that the above displays the amount of utilitie's gathered in each cycle.  
  
sum(v\_tu\_SoC)

## [1] 8.923049

sum(v\_tu\_Exp)

## [1] 17.29923

# Particularly, these are quality adjusted cycles, these are quality adjusted life years where cycles are annual, so I need to consider what this means for utility where cycles are monthly, or fortnightly.  
  
# When I calculate the QALYs above, I don’t convert these quality adjusted cycles to years. If I sum each of v\_tu\_SoC and v\_tu\_Exp I get 16.7 quality adjusted cycles in the SoC arm and 21.1 quality adjusted cycles in the Exp arm. I can convert these quality adjusted cycles to years for fortnights by working out how many fortnights there are in a year (26.0714) and then divide by this number. These correspond to 0.64 and 0.81 QALYs respectively so 0.17 QALYs gained.  
  
v\_tu\_SoC <- v\_tu\_SoC/26.0714  
v\_tu\_Exp <- v\_tu\_Exp/26.0714  
  
# So, these are per-cycle utility values, but, our cycles aren't years, they are fortnights, so these are per fortnight values, if we want this value to reflect the per year value, so that we have a quality adjusted life year, or QALY, then we need to adjust this utility value by how many of these fortnights there are in a year (26.0714), that is divide by how many fortnights there are in a year to bring the per fortnight value to a per year value  
  
  
# James, I'd like your thoughts on what I've done here and whether this seems reasonable.  
  
  
sum(v\_tu\_SoC)

## [1] 0.3422543

sum(v\_tu\_Exp)

## [1] 0.6635328

# QALYS Gained:  
  
Qalys\_gained <- v\_tu\_Exp - v\_tu\_SoC  
sum(Qalys\_gained)

## [1] 0.3212785

# You can see above that there are no utility differences between the different treatments considered: c(u\_H, u\_S, u\_D), it's just different utilities per the health states people are in.  
  
# If we did want to do different utilities for the health state you are in per the treatment you are on, we could define this in the input parameters and then add this in above when creating the vector of utilities for that treatment.  
  
sum(v\_tc\_SoC)

## [1] 25190.26

sum(v\_tc\_Exp)

## [1] 80804.12

# The question is, should I be making that similar conversion of cycles to years for costs? I could do this as below:   
  
# v\_tc\_SoC <- v\_tc\_SoC/26.0714  
# v\_tc\_Exp <- v\_tc\_Exp/26.0714  
#   
# sum(v\_tc\_SoC)  
# sum(v\_tc\_Exp)  
  
  
# But, I assume my costs could be correct, if I correctly defined my costs per cycle.  
  
# This is probably where I would like some feedback, is the manner in which I generate my costs (particularly) and QALYs above, correct?  
  
  
# Reviewing the literature, the fact that: Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., El-Rayes, B. F., & Flowers, C. R. (2015). First-and second-line bevacizumab in addition to chemotherapy for metastatic colorectal cancer: a United States–based cost-effectiveness analysis. Journal of Clinical Oncology, 33(10), 1112. has similar costs, QALYS and Life Years gained and reports similar ICERs (the $352,734/QALY in the UK, which is similar to my own several hundred thousand per QALY without dividing costs by 26.0714 and dissimilar to my ~18,000 per QALY when I divide costs by 26.0714 makes me think I shouldnt be dividing costs by 26.0714).

# There's probably a more elegant way to do this, but if I wanted to add in the once off cost of the COSLOSSUS test, I could do something like the below:  
  
# v\_tc\_trtA <- v\_tc\_trtA+10  
# v\_tc\_trtB <- v\_tc\_trtB+100  
  
# I just need to think about this and does it make sense, because it's adding to the cost each cycle, so probably the best indicator would be whether the cost outputted in 07.2 is only higher by the cost of the test when we do this, or if it is higher by a number of other costs, so I can check that below when adding costs in.  
  
# Page 17 of An Introductory Tutorial on Cohort State-Transition Models in R Using a Cost-Effectiveness Analysis Example https://arxiv.org/pdf/2001.07824.pdf suggests that these are total costs expected per individual, and with testing being a one off cost, maybe you could just add it onto the costs you have at the end? Table 4: Total expected discounted QALYs and costs per average individual in the cohort of the Sick-Sicker model by strategy accounting for within-cycle correction. Costs QALYs Standard of care $151,580 20.711 Strategy A $284,805 21.499 Strategy B $259,100 22.184 Strategy AB $378,875 23.137 The total expected discounted QALYs and costs for the Sick-Sicker model under the four strategies accounting for within-cycle correction are shown in Table 4.

## 07.2 Discounted Mean Costs and QALYs

# Finally, we'll aggregate these costs and utilities into overall discounted mean (average) costs and utilities.  
  
# Obtain the discounted costs and QALYs by multiplying the vectors of total cost and total utility we created above by the discount rate for each cycle:  
  
# Its important to remember what scale I'm on when I applied my discounting formula.  
# If I set d\_e<-0 then my code estimates 16.7 and 21.1 QALYs in each group which must be the quality adjusted cycles.  
  
# Setting the discount rate back to 4% gives me 1.97 and 1.98 QALYs (which are really QA-cycles).  
  
# Looking at the discounting vector I have defined below, I have converted cycles to days but I need to convert the discount rate to a daily discount. If I don't, the result is that discounting reduces the cycles dramatically which reduces the difference which increases with time.  
  
# I can adress this by defining the discount rate as divided by 365 (i.e. the number of days in a year) then the results become 16.4 and 20.6 QA-cycles which of course become 0.63 and 0.79 QALYs respectively, or 0.16 QALYs gained.  
  
d\_c <- d\_c/365  
d\_e <- d\_e/365  
  
# - Then, the discount rate for each cycle needs to be defined accounting for the cycle length, as below:  
  
  
v\_dwc <- 1 / ((1 + d\_c) ^ ((0:(n\_cycle-1)) \* t\_cycle))   
v\_dwe <- 1 / ((1 + d\_e) ^ ((0:(n\_cycle-1)) \* t\_cycle))  
  
  
# So, below we take the vector of costs, transposing it [the t() bit] to make it a 1 row matrix and using matrix multiplication [%\*%] to multiply it by that discount factor vector, which is what you multiply by the outcome in each cycle to get the discounted value of the outcome for that cycle, and then it will all be summed all together across all cycles [across all cells of the 1 row matrix]. Giving you tc\_d\_SoC which is a scalar, or a single value, which is the lifetime expected cost for an average person under standard of care in this cohort.   
  
# Discount costs by multiplying the cost vector with discount weights (v\_dwc)   
# tc\_d\_SoC <- t(v\_tc\_SoC) %\*% v\_dwc  
# tc\_d\_trtA <- t(v\_tc\_trtA) %\*% v\_dwc  
# tc\_d\_trtB <- t(v\_tc\_trtB) %\*% v\_dwc  
  
tc\_d\_SoC <- t(v\_tc\_SoC) %\*% v\_dwc   
tc\_d\_Exp <- t(v\_tc\_Exp) %\*% v\_dwc  
  
  
# So, now we have the average cost per person for treatment with standard of care, and the experimental treatment.   
  
  
# Discount QALYS by multiplying the QALYs vector with discount weights (v\_dwe) [probably utilities would have been a better term here, if I hadnt of updated it from fortnightly health state quality of life, to yearly health state quality of life]  
# tu\_d\_SoC <- t(v\_tu\_SoC) %\*% v\_dwe  
# tu\_d\_trtA <- t(v\_tu\_trtA) %\*% v\_dwe  
# tu\_d\_trtB <- t(v\_tu\_trtB) %\*% v\_dwe  
  
  
tu\_d\_SoC <- t(v\_tu\_SoC) %\*% v\_dwe  
tu\_d\_Exp <- t(v\_tu\_Exp) %\*% v\_dwe  
  
  
# Store them into a vector -> So, we'll take the single values for cost for an average person under standard of care and the experimental treatment and store them in a vector v\_tc\_d:  
v\_tc\_d <- c(tc\_d\_SoC, tc\_d\_Exp)  
v\_tu\_d <- c(tu\_d\_SoC, tu\_d\_Exp)  
  
v\_tc\_d

## [1] 24703.48 78860.21

v\_tu\_d

## [1] 0.3346428 0.6461645

# To make things a little easier to read we might name these values what they are costs for, so we can use the vector of strategy names [v\_names\_str] to name the values:  
  
names (v\_tc\_d) <- v\_names\_strats  
v\_tc\_d

## Standard of Care Experimental Treatment   
## 24703.48 78860.21

names (v\_tu\_d) <- v\_names\_strats  
v\_tu\_d

## Standard of Care Experimental Treatment   
## 0.3346428 0.6461645

Discounted\_Qalys\_gained <- tu\_d\_Exp - tu\_d\_SoC  
sum(Discounted\_Qalys\_gained)

## [1] 0.3115216

# For utility, the utility values aren't different for the different states depending on the treatment strategy, i.e. SOC, Experimental Treatment, but the time spent in the states with the associated utility is different due to the treatment you're on, so your utility value will be higher if the treatment keeps you well for longer so that you stay in a higher utility state for longer than a lower utility state, i.e., progression.  
  
  
# Dataframe with discounted costs and effectiveness  
  
# So then we aggregate them into a dataframe with our discounted costs and utilities, and then we use this to calculate ICERs in: ## 07.3 Compute ICERs of the Markov model  
  
# df\_ce <- data.frame(Strategy = v\_names\_strats,  
# Cost = v\_tc\_d,   
# Effect = v\_tu\_d)  
# df\_ce

## 07.3 Compute ICERs of the Markov model

This study also found that Bevacizumab wasnt a cost-effective addition to care in a number of studies, so I should follow their manner of discussing this: Goldstein, Daniel A., et al. “Bevacizumab for Metastatic Colorectal Cancer: A Global Cost‐Effectiveness Analysis.” The Oncologist 22.6 (2017): 694-699. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5469597/pdf/onco12164.pdf>

I think somewhere in all my example files there must be some example code on generating life years, etc., and reporting them as part of your results table. I basically want to be able to report the following per page 5 of Cost Effectiveness Analysis of Pharmacokinetically-Guided 5-Fluorouracil in FOLFOX Chemotherapy for Metastatic Colorectal Cancer <file:///C:/Users/jonathanbriody/Downloads/goldstein2014.pdf>:

Results In the base case analysis, effectiveness and costs were compared between the PK and BSA group. PK FOLFOX provided 2.03 QALYs at a cost of $50,177 compared with BSA FOLFOX with 1.46 QALYs at a cost of $37,173. The ICER for PK FOLFOX was $22,695 per QALY. In terms of LYs, PK FOLFOX provided 2.60 LYs compared with BSA FOLFOX with 1.99 LYs. The ICER of PK 5-FU was $21,423 per LY. All numerical results are summarized in Table 5.

The code to do this is here to add costs, QALYS, NMBs and Life Years to your leauge table:

cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html

(If I wanted to take a Net Monetary Benefit (NMB) approach instead, I could go to the 35 minute mark of: C:- Royal College of Surgeons in IrelandResources-Effectiveness and Decision Modeling using R Workshop \_ DARTH\_24Session RecordingSession Recording August 24th.mp4)

DOMINATED STRATEGIES EXPLAINED: <https://cran.r-project.org/web/packages/dampack/vignettes/basic_cea.html>

# The discounted costs and QALYs can be summarized and visualized using functions from the 'dampack' package  
(df\_cea <- calculate\_icers(cost = c(tc\_d\_SoC, tc\_d\_Exp),  
 effect = c(tu\_d\_SoC, tu\_d\_Exp),  
 strategies = v\_names\_strats))

## Strategy Cost Effect Inc\_Cost Inc\_Effect ICER Status  
## 1 Standard of Care 24703.48 0.3346428 NA NA NA ND  
## 2 Experimental Treatment 78860.21 0.6461645 54156.73 0.3115216 173845.8 ND

df\_cea

## Strategy Cost Effect Inc\_Cost Inc\_Effect ICER Status  
## 1 Standard of Care 24703.48 0.3346428 NA NA NA ND  
## 2 Experimental Treatment 78860.21 0.6461645 54156.73 0.3115216 173845.8 ND

# df\_cea <- calculate\_icers(cost = df\_ce$Cost,  
# effect = df\_ce$Effect,  
# strategies = df\_ce$Strategy  
# )  
# df\_cea  
  
# The above uses the DARTHtools package to calculate our ICERS, incremental cost and incremental effectiveness, and also describes dominance status:  
  
# This uses the "calculate\_icers function", which does all the sorting, all the prioritization, and then computes the dominance, and not dominance, etc., and there's a publication on the methods behind this, based on a method from colleagues in Stanford.  
  
# The default view is ordered by dominance status (ND = non-dominated, ED = extended/weak dominance, or D= strong dominance), and then ascending by cost per: https://cran.r-project.org/web/packages/dampack/vignettes/basic\_cea.html  
  
  
# The icer object can be easily formatted into a publication quality table using the kableExtra package, as below, but there's probably a better way to do this per Dampack just under:  
  
# library(kableExtra)  
# library(dplyr)  
# df\_cea %>%  
# kable() %>%  
# kable\_styling()  
  
  
## CEA table in proper format ---- per: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\analysis\cSTM\_time\_dep\_simulation.r  
table\_cea <- format\_table\_cea(df\_cea) # Function included in "R/Functions.R"; depends on the `scales` package  
table\_cea

## Strategy Costs ($) QALYs Incremental Costs ($)  
## 1 Standard of Care 24,703 0.33 <NA>  
## 2 Experimental Treatment 78,860 0.65 54,157  
## Incremental QALYs ICER ($/QALY) Status  
## 1 NA <NA> ND  
## 2 0.31 173,846 ND

## Results

I like the way they describe their results: *A Cost-Effectiveness Analysis of Currently Approved Treatments for HBeAg-Positive Chronic Hepatitis B* [*https://sci-hub.ru/10.2165/00019053-200826110-00006*](https://sci-hub.ru/10.2165/00019053-200826110-00006)

A good way to report your results can be seen in Table 4 of the following:

Rivera, F., Valladares, M., Gea, S., & López-Martínez, N. (2017). Cost-effectiveness analysis in the Spanish setting of the PEAK trial of panitumumab plus mFOLFOX6 compared with bevacizumab plus mFOLFOX6 for first-line treatment of patients with wild-type RAS metastatic colorectal cancer. Journal of Medical Economics, 20(6), 574-584. <https://sci-hub.st/10.1080/13696998.2017.1285780> also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Evidence Synthesis\Economic Models\Rivera et al\_2017\_Cost-effectiveness analysis in the Spanish setting of the PEAK trial of.pdf

## 07.4 Plot frontier of the Markov model

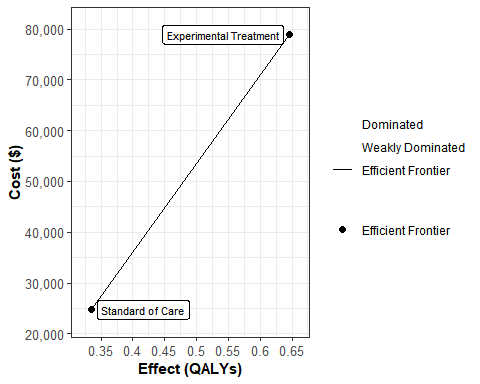
Understand this plot here:

<https://cran.r-project.org/web/packages/dampack/vignettes/basic_cea.html>

Also saved here:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\sensitivity analysis help files\Basic Cost Effectiveness Analysis.pdf

plot(df\_cea, effect\_units = "QALYs", label = "all")



# plot(df\_cea, effect\_units = "QALYs")  
  
  
# When we plot it we have 2 strategies it is possible that something would be off the frontier, would be weakly dominated or strongly dominated, with just a few strategies it's not necessarily that impressive, but with lots of strategies then dampack can be helpful.

The bottom axis of the above diagram shows you how effective the intervention is, the y axis shows you how costly the intervention is, here standard of care is cheaper but less effective than treatment B, which is more effective but more expensive, treatment A is not on the frontier, it has been dominated.

The frontier is also discussed here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R\_HTA\_LMIC\_Intro\_to\_RHTA\_modeling

(If I wanted to see an example where the frontier doesnt exist, I could go to the 38 minute mark of: C:- Royal College of Surgeons in IrelandResources-Effectiveness and Decision Modeling using R Workshop \_ DARTH\_24Session RecordingSession Recording August 24th.mp4)

<https://www.google.ie/search?q=icer+fronter&hl=en&dcr=0&ei=nerzYYS_AciHhbIPlqyMuAk&ved=0ahUKEwjEgar1wdT1AhXIQ0EAHRYWA5cQ4dUDCA4&uact=5&oq=icer+fronter&gs_lcp=Cgdnd3Mtd2l6EAMyBwghEAoQoAE6BwgAEEcQsAM6BQgAEJECOgsIABCABBCxAxCDAToOCC4QgAQQsQMQxwEQowI6CwguELEDEMcBEK8BOg4ILhCABBCxAxDHARDRAzoICAAQsQMQgwE6BQguEJECOgQIABBDOgcIABCxAxBDOggIABCABBCxAzoOCC4QgAQQsQMQxwEQrwE6CAguEIAEELEDOgcIABDJAxBDOgUIABCSAzoFCAAQgAQ6CwguEIAEEMcBENEDOgUILhCABDoLCC4QgAQQxwEQrwE6BAgAEAo6CgguEMcBENEDEAo6BAguEAo6BwgAEMkDEAo6BggAEBYQHjoICAAQFhAKEB46BAgAEA06BQghEKABSgQIQRgASgQIRhgAUKIJWJYZYNEbaANwAngAgAGMAYgB6AiSAQM4LjSYAQCgAQHIAQjAAQE&sclient=gws-wiz>

<https://yhec.co.uk/glossary/cost-effectiveness-frontier/>

[<https://www.hiqa.ie/sites/default/files/2017-01/Revised\_Economic\_Guidelines\_posted\_100714.pdf>](https://www.hiqa.ie/sites/default/files/2017-01/Revised_Economic_Guidelines_posted_100714.pdf) [also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\Revised\_Economic\_Guidelines\_posted\_100714.pdf]

<https://researchonline.lshtm.ac.uk/id/eprint/4648686/1/The%20efficiency-frontier%20approach%20for%20health_GREEN%20AAM.pdf> also saved here:C:- Royal College of Surgeons in IrelandCode\_Modelefficiency-frontier approach for health\_GREEN AAM.pdf

The efficiency frontier is described on page 277 [in the textbook] of: <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/R%20Code/GitHub/COLOSSUS_Model/(Cambridge%20medicine)%20Hunink,%20M.%20G.%20Myriam_Weinstein,%20Milton%20C%20-%20Decision%20making%20in%20health%20and%20medicine_%20integrating%20evidence%20and%20values.pdf>

# Sensitivity Analysis Below:

“It is well known7,8 that conventional univariate sensitivity analysis, whereby individual parameters are varied while maintaining all remaining parameters at their baseline value, is likely to underestimate uncertainty because, in reality, parameters will not vary in isolation. Probabilistic sensitivity analysis involves specifying distributions for model parameters to represent uncertainty in their estimation and employing Monte Carlo simulation to select values at random from those distributions.5,6 In this way, probabilistic models allow the effects of joint uncertainty across all the parameters of the model to be considered. , in probabilistic modeling, parameters are considered random variables, which can take a range of values described by the specified distribution. Parameters in decision models represent summary values related to the average experience across a population of (potential) patients. Therefore, the relevant uncertainty to capture in the formation of a distribution for the parameter is 2nd-order uncertainty related to the sampling distribution of the parameter, not the variability in the values observed in a particular population (1st-order uncertainty; see Stinnett and Paltiel3 for further discussion).” This also has mathematical notation for calculating Beta and Gamma distributions at the end. Probabilistic Analysis of Cost-Effectiveness Models: Choosing between Treatment Strategies for Gastroesophageal Reflux Disease Andrew H. Briggs, DPhil, Ron Goeree, MA, Gord Blackhouse, MBA, Bernie J. O’Brien, PhD www.med.mcgill.ca/epidemiology/courses/EPIB654/Summer2010/EF/example%20PPI.pdfinsu

All plots produced by dampack are ggplot objects and can be readily customized. <https://cran.r-project.org/web/packages/dampack/vignettes/psa_analysis.html>

I’ve put detailed notes in:

C:- Royal College of Surgeons in IrelandResources-Effectiveness and Decision Modeling using R Workshop \_ DARTH\_27\3\_SA\_material\_sick-sicker\_SA\_solutions

Which should be read in tangent with: C:- Royal College of Surgeons in IrelandResourcesModelling - Advanced Course\_Making Models Probabilistic.2 Distributions for parameters.txt

when doing the sensitivity analysis below:

There is similarly relevant information at the 1:38 hour mark of: C:- Royal College of Surgeons in IrelandResources-Effectiveness and Decision Modeling using R Workshop \_ DARTH\_25Session RecordingSession Recording August 25th WITH CHAT.mkv

And in this document:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Briggs et al 2012 model parameter estimation and uncertainty.pdf

**Negative ICER’s:**

In presenting 1-way uncertainty analysis, reporting negative incremental cost-effectiveness ratios (ICERs) should be avoided as they are meaningless.26,27 Instead, the ICER range should be limited to results corresponding to positive incremental health consequences and costs—quadrant I in the cost-effectiveness plane. Results for which incremental costs are positive and health consequences negative should be indicated qualitatively as ‘’dominated’’ and those with negative incremental costs and positive health consequences as ‘’dominant.’’ ICERs corresponding to negative incremental costs and health consequences—quadrant III—should be distinguished from ICERs in quadrant I. C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf

# 08 Deterministic Sensitivity Analysis

A good description of one way sensitivity is here: Cost-effectiveness of capecitabine and bevacizumab maintenance treatment after first-line induction treatment in metastatic colorectal cancer <https://sci-hub.ru/10.1016/j.ejca.2017.01.019>

When i’m thinking about the range around the point estimate, i.e., the max and min, I’ll have to return to my uncertainty notepad file and see how daniel did it to make his table in the goldstein paper because I know he used the same range there for the DSA and PSA based on how he described the results of the tornado diagram, i.e.:

Also I can look at the following study and their Table:

We then assessed the impact of varying single parameters on incremental costs, QALYs, and ICUR. We used the 95% CI of the hazard ratios for death and tumor progression reported in the trial to vary the survival in each health state. We varied utilities in stable disease states from the worst case scenario (cough, dyspnea, and pain) to the best case scenario (treatment response with no symptoms) [15,16]. We varied the utility of PD and FN states from the worst health state utility in our study (severe bleeding) to the utility of stable disease on treatment state [15,17]. We varied the utility of severe bleeding by 10%. We varied the number of bevacizumab cycles from a minimum of 4 to a maximum of 15 cycles, to reflect a range of cycles going from the minimally recommended number of platinum-doublet cycles with no maintenance [4]to continuation of bevacizumab for 3 months after tumor progression [5,15]. We varied all other unit costs by 20% of the base case. Discount rates varied from 0% to 6% (Table 2A). We repeated the one-way sensitivity analysis assuming a dose of 7.5 mg/kg of bevacizumab while varying the other individual parameters within the same ranges used for the dose of 15 mg/kg. This allowed us to estimate the cost-effectiveness and model robustness of bevacizumab at the dose used in the AVAiL trial [22]. Goulart, B., & Ramsey, S. (2011). A trial-based assessment of the cost-utility of bevacizumab and chemotherapy versus chemotherapy alone for advanced non-small cell lung cancer. Value in Health, 14(6), 836-845. sciencedirect.com/science/article/pii/S1098301511014124

We assumed the mean value of each distribution to be the base case value of the corresponding variable, and the standard deviation to be half of the 95% confidence intervals for utility and 20% of the mean values for other variables. So, I can just cite this when I have point estimates and want to build a range around these: Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Ramalingam, S. S., … & Flowers, C. R. (2015). Necitumumab in metastatic squamous cell lung cancer: establishing a value-based cost. JAMA oncology, 1(9), 1293-1300.

[There are more studies that take the 20% approach describe in this file under “the range around things”].

A number of sensitivity analyses were conducted to address uncertainty in parameter estimation and assess robustness of the model results.

A one-way sensitivity analysis evaluates the relative weight of each parameter in our model on overall uncertainty of cost-effectiveness (Hamdy Elsisi et al. 2019).

We varied each model parameter one at a time, while holding the other parameters constant, subsequently we assessed the impact on incremental costs, QALYs, and ICER.

The mean, range and distributions applied in these analyses are described in Table X [big table that has the base-case, max, min, reference and distribution].

Goldstein says “Utilities were varied over their 95% confidence intervals.” so I can think about that.

The below supports my thoughts on the above, that I use the same range for DSA and PSA in the descriptive Table like Goldstein did.

“The general principle remains that assumptions for specifying the distribution and/or defining the interval for uncertainty analysis should follow standard statistical methods (e.g., beta distributions are a natural match for binomial data; gamma or log normal for right skew parameters; log normal for relative risks or hazard ratios; logistic for odds ratios15). These distributions can be used directly in PSA or to define the interval (plausible range) for a DSA.” <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>

“This is true whatever the uncertainty analysis’s technical specification. For a 1-way DSA, it is necessary to specify the parameter’s point estimate and a defensible range; these may be taken directly from the estimation process, with the latter based, for example, on a 95% confidence interval. Representation of uncertainty depends on the uncertainty analysis planned. For DSA, an interval estimate representing beliefs about the parameter’s plausible range is required. For PSA, a distribution is specified via its parameters.” <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>

For cost parameters, a template was completed….

Costs, utilities and probabilities were varied within +/- 20% of baseline values, per (Goldstein et al. 2014b; Goulart and Ramsey 2011b)…. [these are just guys who have done the 20% varying, if I find better online I can use that].

To study the influence of a range of transition risks, the PFS and OS curves $S(t)$ in the model were altered for both treatment strategies via a hazard ratio adjustment $\gamma$, as $[S(t)^{\gamma}]$. Subsequently, transition risks for each cycle were calculated from these adjusted survival curves (Goldstein et al. 2014b). The adjustment hazard ratio was 1 +/- 0.2, i.e., the base case varied within plus or minus 20%.

We individually consider the influence of each parameter on the ICER in a one way deterministic sensitivity analysis.

**Describing Tornado diagram results:**

We create a tornado plot of the one-way sensitivity analysis . In Figure X, the parameters with the largest effect on the ICER differed between treatment arms. Results remained robust to adjustments to X, with an ICER range between $X,000 to X,000 per QALY. There was a limited influence of variation in other parameters on the ICER of THE NOVEL THERAPY (< $X,000 per QALY) . A parameter of interest was X, which - when varied between $X00 and $X00 - placed the ICER range between $X,000 and X,000 per QALY .

[- read this in the context of this article and mirror their display, particularly putting the max and the min on either end of the tornado diagram bars: Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., … & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. Clinical colorectal cancer, 13(4), 219-225.]

Across a range of parameter values, the ICER was above X per QALY, supporting the robustness of results.

[]

“One-way DSA may be reported using a tornado diagram (Figure 1). The horizontal axis is the outcome; along the vertical axis, parameters are arrayed and horizontal bars represent the outcome range associated with the specified parameter’s range. The outcome point estimate corresponding to base-case values is indicated by a vertical line cutting through all horizontal bars. Commonly, the longest bar (reflecting the parameter generating the widest uncertainty) is placed at the top, and the other bars are arrayed in descending order of length. A tornado diagram should be accompanied by a legend or table indicating the upper and lower bounds of values for each parameter, with their justification in terms of the evidence base. A table may be used instead of a tornado diagram or the results ranges provide in the text of the report (e.g., the text might state that ‘’the outcome ranged from X to Y when parameter Z was varied from A to B’’). It is important that the range A to B represents a defensible range for parameter Z, not an arbitrary one.”

Results of 1-way threshold analyses are easily reported in text (e.g., ‘’The ICER remains less than Y as long as the value of X is greater than A,’’ or ‘’Alternative 1 dominates 2 if the value of Z is less than B’’).

<file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>

The tornado plot in Figure X describes which parameters are driving the majority of the variation in the outcome. In the figure, “Parameter Level High/Low” corresponds to parameter values that are above or below the median point estimate. Subsequently, it is possible to highlight where high or low expected outcome values result from variation in parameter values.

*For example, the tornado plot below tells us that the parameter "muDieCancer" has the most leverage in affecting the effectiveness model outcome, and that values below the median value of "muDieCancer" in the one-way sensitivity analysis are associated with higher expected effectiveness outcomes.* [Helpful in interpreting my own tornado diagram with NMB].

*It is important to note that some important information is obscured by tornado plots and caution should be exercised when interpreting it. As the parameter of interest varies across its range in the one-way sensitivity, the strategy that maximizes the outcome of interest can also change across this range. The plot is not showing how the expected outcome changes for a single strategy, but how the expected outcome of the optimal strategy changes. The designation of which strategy is optimal is liable to alternate over the range of the parameter of interest, and this is hidden in a tornado plot.* [Important just generally to note].

*For owsa objects that contain many parameters that have minimal effect on the parameter of interest, you may want to consider producing a plot that highlights only the most influential parameters. Using the min\_rel\_diff argument, you can instruct owsa\_tornado to exclude all parameters that fail to produce a relative change in the outcome below a specific fraction.*

*owsa\_tornado(o,*

*min\_rel\_diff = 0.05)*

*In order to attain the data.frame used to produce the tornado plot, use the return argument to change the type of object returned by the owsa\_tornado function.*

*owsa\_tornado(o,*

*return = “data”)*

*-* [*https://cran.r-project.org/web/packages/dampack/vignettes/psa\_analysis.html*](https://cran.r-project.org/web/packages/dampack/vignettes/psa_analysis.html)

Page 26 of <https://cran.r-project.org/web/packages/dampack/dampack.pdf> has information on making changing to the visual parts of Tornado plots, putting things in black and white, etc.,.

Very simple tornado diagram explanation: <https://mbounthavong.com/blog/2018/5/26/communicating-data-effectively-with-data-visualizations-tornado-diagram>

At the moment, my tornado plot describes a NMB. If I wanted to change this to describe an ICER, it looks like according to the below I could do this just by adding the code from above to generate an ICER to the oncologySemiMarkov\_function.R, storing it in df\_cea, make sure it equals the ICER generated above and change:

outcomes = c(“NMB”), # output to do the OWSA on

to

outcomes = c(“ICER”), # output to do the OWSA on

But, at the moment I have code above where doing a tornado diagram on effectiveness as an outcome is described, so I can take their approach and apply my code to NMB if I like.

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\Deterministic Sensitivity Analysis\_ Generation.pdf

If a reviewer wants me to do a more unusual deterministic sensitivity analysis, which is probabilistic or includes regression, I can find information on doing that here: <https://cran.r-project.org/web/packages/dampack/vignettes/psa_analysis.html>

**Describing plotted outcome values over a range of values of the parameters of interest and treatment strategies:** OR, MORE SIMPLY PUT -> **Describing Plotting One Way Sensitivity Analysis (OWSA) Plot:**

In Figure X we conduct a one-way deterministic sensitivity analysis separately on each of the different parameters, per treatment strategy. This figure illustrates how the expected outcome value from the decision model varies (on the x axis) as the parameters vary across the range of minimum to maximum values described in Table X (on the y axis).

**Describing Optimal Strategy:**

While a tornado plot provides a useful visual aid to identify which parameters are driving the majority of the variation in the outcome of interest, such plots also obscure the fact that as a parameter changes from it’s minimum to it’s maximum value, the treatment strategy that maximises the specified model outcome may also vary across this range. Tornado plots depict changes in the expected outcome of the optimal strategy, rather than for a single strategy. However, this “optimal strategy” can vary when the parameter of interest alternates from low to high values, something which is generally obsfucated in tornado plots. Consequently, we also conduct a visualization of the optimal strategy, i.e., the strategy that maximises the expected outcome, over each parameter range in Figure X. We derive this optimal strategy plot from the results of the one-way sensitvity analysis. We only plot parameters that make changes in the optimal strategy as they vary. Thus, we can illustrate when the treatment strategy (SoC or experimental treatment) maximising the expectation of the outcome of interest (here NMB, but we can change this to ICERS) alternates as a function of each parameter of interest.

[Per: [https://cran.r-project.org/web/packages/dampack/vignettes/psa\_analysis.html]](https://cran.r-project.org/web/packages/dampack/vignettes/psa_analysis.html%5D)

[If I do this and my optimal strategy says that the designation of which strategy is optimal doesnt alternate, I can report this as a result and show all my non-altered optimal strategy boxes in the diagram by removing plot\_const = FALSE, rather than thinking it invalidates the reason for doing an optimal strategy plot].

In plotting the optimal strategy as the parameter values change, in Figure X we demonstrate that the optimal strategy varies from SoC to Exp as X PARAMETER changes from Y to Z.

**Describing Two-Way Sensitivity Analysis:**

Finally, we *conduct a two-way uncertainty analysis to* investigate changes *in the outcome* when varying two parameters simultaneously. As informed by (Briggs et al. 2012) we conduct a two-way sensitivity *uncertainty* analysis for parameters expected to have a logical covariance. We examine how the expected outcome (here NMB) changes as both the hazard ratio adjustment applied to the PFS curve under standard of care - and accordingly the event rate in the control intervention - and the hazard ratio with the novel therapy change. The results of this uncertainty analysis are reported in Figure X. The axes represent the range of possible values of the parameters of interest. The quadrant is partitioned into regions according to which treatment strategy maximizes the expected outcome as a function of the two parameters being varied. With ICERS, all the possible ICERS following the TWSA are plotted in the quadrant and the boundary line indicates the specified ICER threshold [i.e., the WTP], where on on each side one treatment is preferred over the other, i.e., above the line treatment strategy A may be preferred to treatment strategy B, and vice-versa below, so, it dominates the other because it’s ICER value is sufficient to make this treatment cost-effective instead of the other.

ICERs explained on page 8 of:

[<file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>](file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf)

Per the following links, you can plot these on any model outcome - such as cost and effect, not just NMB. This is supported by when I went into the ispor webinar and applied these plots with ICERS as the outcomes rather than NMB, and it worked.

<https://cran.r-project.org/web/packages/dampack/vignettes/psa_analysis.html>

syzoekao.github.io/CEAutil/

Model Parameter Estimation and Uncertainty Analysis: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force Working Group–6 Andrew H. Briggs, DPhil, Milton C. Weinstein, PhD, Elisabeth A. L. Fenwick, PhD, Jonathan Karnon, PhD, Mark J. Sculpher, PhD, A. David Paltiel, PhD, on Behalf of the ISPOR-SMDM Modeling Good Research Practices Task Force

Figure 1 here: <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>

Cost Effectiveness Analysis of Pharmacokinetically-Guided 5-Fluorouracil in FOLFOX Chemotherapy for Metastatic Colorectal Cancer Daniel A. Goldstein,1 Qiushi Chen,2 Turgay Ayer,2 David H. Howard,1,3 Joseph Lipscomb,1,3 R. Donald Harvey,1 Bassel F. El-Rayes,1 Christopher R. Flowers

<file:///C:/Users/jonathanbriody/Downloads/goldstein2014.pdf>

Cost-effectiveness analysis of fruquintinib versus regorafenib as the third-line therapy for metastatic colorectal cancer in China Xin Guan, Hongchao Li, Xiaomo Xiong, Cike Peng, Ning Wang, Xiao Ma & Aixia Ma

<https://www.tandfonline.com/doi/pdf/10.1080/13696998.2021.1888743>

Cost-effectiveness of sorafenib versus best supportive care in advanced hepatocellular carcinoma in Egypt Gihan Hamdy Elsisi, Yousery Nada, Noha Rashad & João Carapinha <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Evidence%20Synthesis/Paper%20Materials%20and%20Methods/Hamdy%20Elsisi%20et%20al_2019_Cost-effectiveness%20of%20sorafenib%20versus%20best%20supportive%20care%20in%20advanced.pdf>

A Trial-Based Assessment of the Cost-Utility of Bevacizumab and Chemotherapy versus Chemotherapy Alone for Advanced Non-Small Cell Lung Cancer Bernardo Goulart, MD, MS1,2,\*, Scott Ramsey, MD, PhD1, <https://sci-hub.ru/10.1016/j.jval.2011.04.004>

## 08.1 Load PFS-PFSer Markov model function

To make the function work for my health states I rename “health” and “sick” in it to “PFS” and “OS” respectively.

One thing that concerns me is that the first line of the function sets willingness to pay at 10,000 and this is then used to build the net monetary benefit, I wonder if this needs to be updated for different willingess to pay values directly in the function or if I can do it in this Markov\_3state.Rmd file?

oncologySemiMarkov <- function(l\_params\_all, n\_wtp = 10000) {

It may just be creating an wtp for the NMB calculation in the function, because there is none for use in this function and one is needed to calculate:

v\_nmb\_d <- v\_tu\_d \* n\_wtp - v\_tc\_d

but probably a good idea to ensure this is the case (i.e., change the n\_wtp to see how it changes the nmb) and also ensure that I am happy with the NMB value currently under use (10,000 above), i.e. do a little reading and see if another n\_wtp may be more appropriate and maybe reach out to darth and see if there is a reason their wtp’s don’t match.

## 08.3 One-way sensitivity analysis (OWSA)

# A brief note on how parameters are varied one at a time:  
  
# A simple one-way DSA starts with choosing a model parameter to be investigated. Next, the modeler specifies a range for this parameter and the number of evenly spaced points along this range at which to evaluate model outcomes. The model is then run for each element of the vector of parameter values by setting the parameter of interest to the value, holding all other model parameters at their default base case values.  
  
# To conduct the one-way DSA, we then call the function run\_owsa\_det with my\_owsa\_params\_range, specifying the parameters to be varied and over which ranges, and my\_params\_basecase as the fixed parameter values to be used in the model when a parameter is not being explicitly varied in the one-way sensitivity analysis.  
  
# [In case I struggle to describe varying the hazard ratio, I describe this below]:  
  
# In the example below, we will conduct four one-way DSAs separately on four different parameters: the utility benefit of Treatment\_A; the cost of Treatment\_A; the reduction, as a hazard ratio, on the rate of progressing from Sick to Sicker with Treatment\_B; and the rate of dying from the Healthy state.  
  
# my\_owsa\_params\_range <- data.frame(pars = c("u\_trtA", "c\_trtA", "hr\_S1S2\_trtB", "r\_HD"),  
# min = c(0.9, 9000, 0.3, 0.001),  
# max = c(1, 24000, 0.9, 0.003))]  
  
# https://cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
  
  
  
# I've written up an explanation of tornado diagram probabilities and the rules behind the creating of boxes in Tornado diagrams, etc., here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\Tornado Diagram Probabilities Explained.R  
  
  
# Using ICERs in the tornado diagram:  
  
# Typically, it's very easy to get the tornado plot to work with ICERS rather than with NMB, you just change outcomes = c("NMB"), to outcomes = c("DSAICER"), and then change the function to include the following code, basically creating ICERs the same way you do in this Rmd file and then an insertion of DSAICER = DSA\_ICER into the df\_ce so that you have an ICER to include under outcomes = c("") as above:   
  
 # v\_nmb\_d <- v\_tu\_d \* n\_wtp - v\_tc\_d  
 #   
 # (df\_DSAcea <- calculate\_icers(cost = c(tc\_d\_SoC, tc\_d\_Exp),  
 # effect = c(tu\_d\_SoC, tu\_d\_Exp),  
 # strategies = v\_names\_strats))  
 # df\_DSAcea  
 #   
 # DSA\_ICER <- c(df\_DSAcea[2,6])  
 #   
 #   
 # df\_ce <- data.frame(Strategy = v\_names\_strats,  
 # Cost = v\_tc\_d,  
 # Effect = v\_tu\_d,  
 # NMB = v\_nmb\_d,  
 # DSAICER = DSA\_ICER)  
 #   
 # return(df\_ce)  
  
  
# However, I was having some trouble getting my tornado plot to generate. I think I've tracked the issue down to ggplot, which the dampack tornado plot code uses. I suspect that this was because some of my values were very small, because the exact issue I was getting was:  
  
#Error in if (zero\_range(as.numeric(limits))) { :  
#missing value where TRUE/FALSE needed   
  
# and per the following webpage https://stackoverflow.com/questions/12462479/r-ggplot2-simple-plot-cannot-specify-log-axis-limits other people have had the same issue and talked about ggplot and how this error might mean it won't accept certain issues. Also here: https://stackoverflow.com/questions/28474630/missing-value-where-true-false-needed-error-in-if-statement-in-r and here:https://stackoverflow.com/questions/55867994/ggplot-has-error-missing-value-where-true-false-needed and here: https://stackoverflow.com/questions/67810309/pot-histograms-and-had-error-missing-value-where-true-false-needed and here: https://stackoverflow.com/questions/53052874/define-new-scales-axis-tranform-for-ggplot and https://github.com/tidyverse/ggplot2/issues/2907 and https://github.com/tidyverse/ggplot2/issues/930 and https://community.rstudio.com/t/can-i-transform-scales-axes-from-log-to-linear-scale-in-ggplot2/69628/7 and https://groups.google.com/g/ggplot2/c/Z5dt6ZBnGII?pli=1  
  
# I should check if they build tornado diagrams without ggplot in the York code, and if they do at least I can see the ggplot in action and remedy the issue using one of the solutions suggested at the above links.  
  
  
# If I can't figure out how to get a tornado diagram to work with ICER's rather than NMB, I could follow the approach here of doing a probabilistic analysis and then creating another tornado diagram with ICERS on the probabilistic analysis. These examples explain the technicalities of how this is done in R but also from a theoretical viewpoint how this works:  
  
# https://github.com/DARTH-git/dampack/blob/master/R/owsa.R  
  
# Also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\doingtornadoprobabilsticallyexample.R  
  
# https://cran.r-project.org/web/packages/dampack/vignettes/psa\_analysis.html also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\cran-r-project-org-web-packages-dampack-vignettes-psa\_analysis-html.pdf  
  
# Alternatively, I could take the following approach described here:   
  
# https://rpubs.com/mbounthavong/decision\_tree\_model\_tutorial also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\rpubs-com-mbounthavong-decision\_tree\_model\_tutorial.pdf with the code that it calls saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\R Code\GitHub\COLOSSUS\_Model\tornado\_diagram\_code.R  
  
# I've actually coded up the above solution in quite a bit of detail. To get it working though I'll have to change oncologySemiMarkov\_function.R to be:  
  
#   
# v\_nmb\_d <- v\_tu\_d \* n\_wtp - v\_tc\_d  
#   
# (df\_DSAcea <- calculate\_icers(cost = c(tc\_d\_SoC, tc\_d\_Exp),  
# effect = c(tu\_d\_SoC, tu\_d\_Exp),  
# strategies = v\_names\_strats))  
# df\_DSAcea  
#   
# DSA\_ICER <- c(df\_DSAcea[2,6])  
#   
# # I'm picking the row and column where the ICER value appears in the df\_DSAcea dataframe created by calculate\_icers.  
#   
# # Generate the ouput  
# return(c(v\_names\_strats, v\_tc\_d, v\_tu\_d, DSA\_ICER))  
# })  
#   
# }  
#   
   
   
# Or I could try the below:  
  
# https://cran.r-project.org/web/packages/tornado/vignettes/tornadoVignette.html also saved here: https://cran.r-project.org/web/packages/tornado/vignettes/tornadoVignette.html  
  
# But if these don't work, because of using ggplot also, I could see if I can find a package or manual creation code that doesnt use ggplot, or seeing as the ggplot part of the code in the mbounthavong code happens somewhere I can see it and change it, I might be able to see if one of the fixes mentioned in the stack overflow links about can be applied and let the tornado plot be built.  
  
# Worst case scenario, I create the tornado diagram and two-way sensitivity analysis as NMB and report NMB's in the league table of results so it doesnt look too out of place.  
  
  
  
#Normal chrome Session buddy has all the tabs on describing tornado diagrams that I need to read saved. Canary session buddy has all the tabs saved on doing a tornado diagram outside of the darth framework.  
  
# A CODE CHUNK WHERE IT IS EASY TO RUN ALL THE CODE IN ONE FILE, WHICH WILL BE TAKEN AND ADDED TO THE ACTUAL TORNADO CODE CHUNKS LOWER DOWN.  
  
  
#p\_PD <- 0.05  
#p\_FD <- 0.02   
#u\_F <- 0.5  
#p\_FD\_SoC <- 0.05  
#p\_FD\_Exp <- 0.05  
  
#p\_FA1\_SoC <- p\_FA1\_STD  
#p\_FA2\_SoC <- p\_FA2\_STD  
#p\_FA3\_SoC <- p\_FA3\_STD  
  
# Because of how p\_FA1\_SoC is made, [i.e., p\_FA1\_SoC <- (p\_PFS\_SoC) \* p\_FA1\_STD means that I will get a different value of p\_FA1\_SoC for each cycle based on multiplying a static value of p\_FA1\_STD by the varying value of p\_PFS\_SoC] I will get the following error if I include it as it was created. I can address this just like the hazard ratio problem, but at the moment I havent thought about probabilities and the whole (p\_PFS\_SoC) \* p\_FA1\_STD part too deeply, so instead, I decide to make it non-varying by including just the static value that is multiplied by the changing p\_PFS\_SoC value as above. This all also holds for p\_FA2\_SoC and p\_FA3\_SoC.  
  
  
# Error in data.frame(pars = c("HR\_FP\_Exp", "HR\_FP\_SoC", "p\_PD", "p\_FD\_SoC", :   
# arguments imply differing number of rows: 29, 386  
  
  
# p\_A1F\_SoC = 0.01  
  
  
# p\_A1D\_SoC <- 0.00100000000000000000  
  
  
# I was having some trouble with the A1D probabilities - with the basecase not actually being between the min and the max. My assumptions was that because the numbers were so small they were being rounded and thus moved to a value that wasnt bigger or smaller than the base case value. Although I don't think this should fix things, as this is just the format in which things are printed to the screen, rather than the value that is saved in the parameter, I still set R to display as many digits as possible (22 is the max) per: https://stackoverflow.com/questions/6669681/preventing-r-from-rounding, as suspected things are unchanged whether options(digits=2) of options(digits=22)  
  
# I realised that the issue was that when the parameter was set equal to a value in the list below, this only changed the parameter value in the list, not outside the list, so when I create my min and max using that parameter, it is using it's initial value, not the value assigned in the list, the problem with this is that if the parameter value in the list and the value in the min and max calculation don't align, then the basecase value of the parameter from the list is unlikely to lie between the min and max created from this parameter value outside the list. I have a github commit where I get into this in greater detail here: https://github.com/JonathanBriody/COLOSSUS\_Model/commit/3b0cac700598e8d111c96ab39e13ce656ae80210 if the link doesnt work it's a commit called "the problem was this" around 5.15pm on the 20/07/22  
  
options(digits=4)  
  
## 4.1 Initialization ----  
  
# Load the model as a function that is defined in the supporting script  
# source("Functions\_markov\_3state.R")  
# Test function  
# calculate\_ce\_out(l\_params\_all)  
  
source(file = "oncologySemiMarkov\_function.R")  
# If I change any code in this main model code file, I will also need to update the function that I call.  
  
  
# Create list l\_params\_all with all input probabilities, costs, utilities, etc.,  
  
# Test whether the function works (and generates the same results)  
# - to do so, first a list of parameter values needs to be generated  
  
# In Koen's code they manually set the value of things here, i.e. HR\_FP\_Exp = 0.6, however, to support replicability I just set the variables equal to themselves for this list, and that way if I change something like HR\_FP\_Exp = 0.6, to HR\_FP\_Exp = 0.9 earlier in this R code it will automatically update here as well.  
  
# I know that the values they used in this section were the same as their earlier values as they gave identical results when changed from manual values to setting the variable names equal to themselves, apart from .01 of a difference for NMB, but that's probably a rounding thing:  
  
  
  
#   
# > l\_params\_all <- list(  
# + coef\_weibull\_shape\_SoC = 0.7700246,  
# + coef\_weibull\_scale\_SoC = 1.742534,  
# + HR\_FP\_Exp = 0.6,  
# + p\_FD = 0.02,   
# + p\_PD = 0.1,   
# + c\_F\_SoC = 400,   
# + c\_F\_Exp = 800,  
# + c\_P = 1000,   
# + c\_D = 0,   
# + u\_F = 0.8,   
# + u\_P = 0.5,   
# + u\_D = 0,   
# + d\_e = 0.03,   
# + d\_c = 0.03,  
# + n\_cycle = 60,  
# + t\_cycle = 0.25  
# + )  
# >   
# > oncologySemiMarkov(l\_params\_all = l\_params\_all, n\_wtp = 20000)  
# Strategy Cost Effect NMB  
# 1 Standard of Care 11493.14 3.725848 63023.81  
# 2 Experimental 18913.35 4.148299 64052.63  
# > l\_params\_all <- list(  
# + coef\_weibull\_shape\_SoC = coef\_weibull\_shape\_SoC,  
# + coef\_weibull\_scale\_SoC = coef\_weibull\_scale\_SoC,  
# + HR\_FP\_Exp = HR\_FP\_Exp,  
# + p\_FD = p\_FD,   
# + p\_PD = p\_PD,   
# + c\_F\_SoC = c\_F\_SoC,   
# + c\_F\_Exp = c\_F\_Exp,  
# + c\_P = c\_P,   
# + c\_D = c\_D,   
# + u\_F = u\_F,   
# + u\_P = u\_P,   
# + u\_D = u\_D,   
# + d\_e = d\_e,   
# + d\_c = d\_c,  
# + n\_cycle = n\_cycle,  
# + t\_cycle = t\_cycle  
# + )  
# > oncologySemiMarkov(l\_params\_all = l\_params\_all, n\_wtp = 20000)  
# Strategy Cost Effect NMB  
# 1 Standard of Care 11493.14 3.725848 63023.82  
# 2 Experimental 18913.35 4.148299 64052.64  
  
# Now I update this list with the variables I have:  
  
# If I updated utility for AE above, then I'll have to take that into account for u\_F below:  
  
l\_params\_all <- list(  
 HR\_FP\_Exp = HR\_FP\_Exp,  
 HR\_FP\_SoC = HR\_FP\_SoC,  
 HR\_PD\_Exp = HR\_PD\_Exp,  
 HR\_PD\_SoC = HR\_PD\_SoC,  
 P\_OSD\_SoC = P\_OSD\_SoC,   
 P\_OSD\_Exp = P\_OSD\_Exp,  
 p\_FA1\_STD = p\_FA1\_STD,  
 p\_FA2\_STD = p\_FA2\_STD,  
 p\_FA3\_STD = p\_FA3\_STD,  
 p\_FA1\_EXPR = p\_FA1\_EXPR,  
 p\_FA2\_EXPR = p\_FA2\_EXPR,  
 p\_FA3\_EXPR = p\_FA3\_EXPR,  
 administration\_cost = administration\_cost,  
 #ITS FINE TO INCLUDE THE INGRIDENIENTS THAT MAKE UP c\_F\_SoC, c\_F\_Exp, c\_P, but don't include c\_F\_SoC, c\_F\_Exp, c\_P themsleves, BECAUSE WE ARE CHANGING WHAT BUILDS THESE COSTS WITH THE INGRIEDIENTS, SO WE DON'T WANT TO CHANGE IT AGAIN HERE ONCE WE'VE CHANGED WHAT BUILDS IT  
 c\_PFS\_Folfox = c\_PFS\_Folfox,  
 c\_PFS\_Bevacizumab = c\_PFS\_Bevacizumab,  
 c\_OS\_Folfiri = c\_OS\_Folfiri,  
 c\_D = c\_D,   
 c\_AE1 = c\_AE1,  
 c\_AE2 = c\_AE2,  
 c\_AE3 = c\_AE3,  
 u\_F = u\_F,   
 #ITS FINE TO INCLUDE U\_F BUT DON'T INCLUDE U\_F\_SoC, BECAUSE WE ARE CHANGING WHAT BUILDS U\_F\_SoC WTH U\_F AND AE1\_DisUtil SO WE DON'T WANT TO CHANGE IT AGAIN HERE ONCE WE'VE CHANGED WHAT BUILDS IT  
 u\_P = u\_P,   
 u\_D = u\_D,   
 AE1\_DisUtil = AE1\_DisUtil,  
 AE2\_DisUtil = AE2\_DisUtil,  
 AE3\_DisUtil = AE3\_DisUtil,  
 d\_e = d\_e,   
 d\_c = d\_c,  
 n\_cycle = n\_cycle,  
 t\_cycle = t\_cycle  
)  
  
   
 #######################################################################  
   
# p\_FA1\_SoC <- (p\_PFS\_SoC) \* p\_FA1\_STD  
# p\_FA2\_SoC <- (p\_PFS\_SoC) \* p\_FA2\_STD  
# p\_FA3\_SoC <- (p\_PFS\_SoC) \* p\_FA3\_STD  
   
 # I think that, to ensure the probability is a conditional probability for p\_FA1\_SoC, I need to define it as above, because I know that p\_PFS\_SoC changes due to the hazard ratio changes, BUT, if I want to alter the probability of the adverse event in my model, I should let: p\_FA1\_STD reflect my adverse event and include this in the OWSA code, rather than p\_FA1\_SoC - because p\_FA1\_STD can be changed by 20% and then this code above will create p\_FA1\_SoC conditional on the new probability for p\_PFS\_SoC as this changes with the changing hazard ratios in the senstivity analysis. So, p\_FA1\_SoC will be conditional on the updated p\_PFS\_SoC rather than the original p\_PFS\_SoC pre-hazard ratio changes to p\_PFS\_SoC.  
  
# The above was applicable when adverse events represented health states initially.  
  
  
  
# Test function  
  
# Test whether the function works (and generates the same results)  
  
oncologySemiMarkov(l\_params\_all = l\_params\_all, n\_wtp = 45000)

## Strategy Cost Effect DSAICER  
## 1 Standard of Care 24703 0.3346 173846  
## 2 Experimental Treatment 78860 0.6462 173846

# I can't use these probabilities because they are from Free to Progressed, but now those probabilities are time-sensitive, so if I try to change these here and include them in the tornado diagram I'll be including too many things, as I explain in my description on hazard ratios.  
  
# Minimum\_p\_FP\_SoC <- p\_FP\_SoC - 0.20\*p\_FP\_SoC  
# Maximum\_p\_FP\_SoC <- p\_FP\_SoC + 0.20\*p\_FP\_SoC  
#   
#   
# Minimum\_p\_FP\_Exp <- p\_FP\_Exp - 0.20\*p\_FP\_Exp  
# Maximum\_p\_FP\_Exp <- p\_FP\_Exp + 0.20\*p\_FP\_Exp  
  
  
# Hazard Ratios:  
  
# To vary transition probabilities is slightly complex in a time dependent model.  
  
  
# FOR THE TRANSITION PROBABILITY FROM FREE TO PROGRESSION WE CAN MULTIPLY THAT PARAMETER BY 20% ONCE AND IT WILL CHANGE THE TRANSITION PROBABILITIES AT EACH CYCLE   
   
# > Minimum\_p\_FP\_SoC <- p\_FP\_SoC - 0.20\*p\_FP\_SoC  
# > Maximum\_p\_FP\_SoC <- p\_FP\_SoC + 0.20\*p\_FP\_SoC  
  
# > p\_FP\_SoC  
# ...0.249399924  
# > Minimum\_p\_FP\_SoC  
# ...0.1995199390  
# > Maximum\_p\_FP\_SoC  
# ...0.299279908  
  
# 0.0499 is 20% of 0.249399924  
# 0.249399924 - 0.0499 = 0.199499924  
# 0.249399924 + 0.0499 = 0.299299924   
  
# So, this allows us to have the 20% variability on our time dependent transition probabilities  
  
  
# WE DONT NEED TO HAVE A MAX (MEAN+20%) AND MIN (MEAN-20%) FOR THE HAZARD RATIO TOO IF WE HAVE THESE FOR TRANSITION PROBABILITIES FOR THE EXPERIMENTAL TREATMENT.  
  
# There's a problem below, df\_params\_OWSA doesnt like the fact that a different probability for each cycle (from the time-dependent transition probabilities) gives 122 rows.   
   
# > df\_params\_OWSA <- data.frame(  
# + pars = c("c\_F\_Exp", "u\_F", "p\_FP\_SoC", "p\_FP\_Exp"), # names of the parameters to be changed  
# + min = c(400, 0.70, Minimum\_p\_FP\_SoC, Minimum\_p\_FP\_Exp), # min parameter values  
# + max = c(1200, 0.90, Maximum\_p\_FP\_SoC, Maximum\_p\_FP\_Exp) # max parameter values  
# + )  
# Error in data.frame(pars = c("c\_F\_Exp", "u\_F", "p\_FP\_SoC", "p\_FP\_Exp"), :   
# arguments imply differing number of rows: 4, 122  
#   
  
# To address this, I can go back to the Goldstein approach and change the hazard ratio instead. For the experimental strategy this is simple to do as there already is a hazard ratio in generating experimental strategy transition probabilities.   
  
# For the standard of care strategy I'll have to add a hazard ratio into oncologySemiMarkov\_function that is equal to 1 and get's multiplied by the standard of care stuff first, then I can vary this by a min and a max as above.  
  
  
  
  
UpperCI <- 0.87  
LowerCI <- 0.53  
  
  
  
  
HR\_FP\_Exp

## [1] 0.68

Minimum\_HR\_FP\_Exp <- LowerCI  
Maximum\_HR\_FP\_Exp <- UpperCI  
  
  
  
HR\_FP\_SoC

## [1] 1

Minimum\_HR\_FP\_SoC <- HR\_FP\_SoC - 0.20\*HR\_FP\_SoC  
Maximum\_HR\_FP\_SoC <- HR\_FP\_SoC + 0.20\*HR\_FP\_SoC  
  
  
# Now that we're using the OS curves, I add hazard ratios for PFS to dead that reflect the hazard ratio of the experimental strategy changing the probability of going from PFS to Death, and the hazard ratio of 1 that apply in standard of care so that I can vary transition probabilities under standard of care in this one-way sensitivity analysis:  
  
HR\_PD\_SoC

## [1] 1

Minimum\_HR\_PD\_SoC <- HR\_PD\_SoC - 0.20\*HR\_PD\_SoC  
Maximum\_HR\_PD\_SoC <- HR\_PD\_SoC + 0.20\*HR\_PD\_SoC  
  
  
OS\_UpperCI <- 0.86  
OS\_LowerCI <- 0.49  
   
   
HR\_PD\_Exp

## [1] 0.65

Minimum\_HR\_PD\_Exp <- OS\_LowerCI  
Maximum\_HR\_PD\_Exp <- OS\_UpperCI  
  
  
  
  
  
  
  
  
# Probability of progressive disease to death:  
  
# Under the assumption that everyone will get the same second line therapy, I give them all the same probability of going from progessed (i.e., OS) to dead, and thus only need to include p\_PD here once - because it is applied in oncologySemiMarkov\_function.R for both SoC and Exp. ACTUALLY I THINK IT SHOULD BE P\_OSD\_SoC P\_OSD\_Exp BOTH INCLUDED.  
  
P\_OSD\_SoC

## [1] 0.17

Minimum\_P\_OSD\_SoC <- 0.12  
Maximum\_P\_OSD\_SoC <- 0.22  
  
P\_OSD\_Exp

## [1] 0.17

Minimum\_P\_OSD\_Exp <- 0.12  
Maximum\_P\_OSD\_Exp <- 0.22  
  
  
  
  
  
  
  
# Probability of going from PFS to Death states under the standard of care treatment and the experimental treatment:  
  
# # HR\_PD\_SoC and HR\_PD\_Exp address this as above:  
  
# p\_FD\_SoC  
#   
# Minimum\_p\_FD\_SoC <- p\_FD\_SoC - 0.20\*p\_FD\_SoC  
# Maximum\_p\_FD\_SoC <- p\_FD\_SoC + 0.20\*p\_FD\_SoC  
#   
# p\_FD\_Exp  
#   
# Minimum\_p\_FD\_Exp<- p\_FD\_Exp - 0.20\*p\_FD\_Exp  
# Maximum\_p\_FD\_Exp <- p\_FD\_Exp + 0.20\*p\_FD\_Exp  
  
  
  
# Probability of Adverse Events:  
  
p\_FA1\_STD

## [1] 0.04

Minimum\_p\_FA1\_STD <- p\_FA1\_STD - 0.20\*p\_FA1\_STD  
Maximum\_p\_FA1\_STD <- p\_FA1\_STD + 0.20\*p\_FA1\_STD  
  
p\_FA2\_STD

## [1] 0.31

Minimum\_p\_FA2\_STD <- p\_FA2\_STD - 0.20\*p\_FA2\_STD  
Maximum\_p\_FA2\_STD <- p\_FA2\_STD + 0.20\*p\_FA2\_STD  
  
p\_FA3\_STD

## [1] 0.31

Minimum\_p\_FA3\_STD <- p\_FA3\_STD - 0.20\*p\_FA3\_STD  
Maximum\_p\_FA3\_STD <- p\_FA3\_STD + 0.20\*p\_FA3\_STD  
  
  
p\_FA1\_EXPR

## [1] 0.07

Minimum\_p\_FA1\_EXPR <- p\_FA1\_EXPR - 0.20\*p\_FA1\_EXPR  
Maximum\_p\_FA1\_EXPR <- p\_FA1\_EXPR + 0.20\*p\_FA1\_EXPR  
  
p\_FA2\_EXPR

## [1] 0.11

Minimum\_p\_FA2\_EXPR <- p\_FA2\_EXPR - 0.20\*p\_FA2\_EXPR  
Maximum\_p\_FA2\_EXPR <- p\_FA2\_EXPR + 0.20\*p\_FA2\_EXPR  
  
p\_FA3\_EXPR

## [1] 0.07

Minimum\_p\_FA3\_EXPR <- p\_FA3\_EXPR - 0.20\*p\_FA3\_EXPR  
Maximum\_p\_FA3\_EXPR <- p\_FA3\_EXPR + 0.20\*p\_FA3\_EXPR  
  
  
  
  
  
  
#install.packages("Rmpfr")  
#library(Rmpfr)  
  
#   
# Minimum\_p\_A1D\_SoC <- mpfr(Minimum\_p\_A1D\_SoC,200) # set arbitrary precision that's greater than R default  
# Maximum\_p\_A1D\_SoC <- mpfr(Maximum\_p\_A1D\_SoC,200) # set arbitrary precision that's greater than R default  
#   
#   
# Minimum\_p\_A2D\_SoC <- mpfr(Minimum\_p\_A2D\_SoC,200) # set arbitrary precision that's greater than R default  
# Maximum\_p\_A2D\_SoC <- mpfr(Maximum\_p\_A2D\_SoC,200) # set arbitrary precision that's greater than R default  
#   
# Minimum\_p\_A3D\_SoC <- mpfr(Minimum\_p\_A3D\_SoC,200) # set arbitrary precision that's greater than R default  
# Maximum\_p\_A3D\_SoC <- mpfr(Maximum\_p\_A3D\_SoC,200) # set arbitrary precision that's greater than R default  
  
  
  
  
# Cost:  
  
# If I decide to include the cost of the test for patients I will also need to include this in the sensitivity analysis here:  
  
administration\_cost

## [1] 365

Minimum\_administration\_cost <- administration\_cost - 0.20\*administration\_cost  
Maximum\_administration\_cost <- administration\_cost + 0.20\*administration\_cost  
  
c\_PFS\_Folfox

## [1] 307.8

Minimum\_c\_PFS\_Folfox <- c\_PFS\_Folfox - 0.20\*c\_PFS\_Folfox  
Maximum\_c\_PFS\_Folfox <- c\_PFS\_Folfox + 0.20\*c\_PFS\_Folfox  
  
c\_PFS\_Bevacizumab

## [1] 2580

Minimum\_c\_PFS\_Bevacizumab <- c\_PFS\_Bevacizumab - 0.20\*c\_PFS\_Bevacizumab  
Maximum\_c\_PFS\_Bevacizumab <- c\_PFS\_Bevacizumab + 0.20\*c\_PFS\_Bevacizumab  
  
c\_OS\_Folfiri

## [1] 326

Minimum\_c\_OS\_Folfiri <- c\_OS\_Folfiri - 0.20\*c\_OS\_Folfiri  
Maximum\_c\_OS\_Folfiri <- c\_OS\_Folfiri + 0.20\*c\_OS\_Folfiri  
  
c\_D

## [1] 0

Minimum\_c\_D <- c\_D - 0.20\*c\_D  
Maximum\_c\_D <- c\_D + 0.20\*c\_D  
  
c\_AE1

## [1] 2836

Minimum\_c\_AE1 <- c\_AE1 - 0.20\*c\_AE1  
Maximum\_c\_AE1 <- c\_AE1 + 0.20\*c\_AE1  
  
c\_AE2

## [1] 1459

Minimum\_c\_AE2 <- c\_AE2 - 0.20\*c\_AE2  
Maximum\_c\_AE2 <- c\_AE2 + 0.20\*c\_AE2  
  
c\_AE3

## [1] 409

Minimum\_c\_AE3 <- c\_AE3 - 0.20\*c\_AE3  
Maximum\_c\_AE3 <- c\_AE3 + 0.20\*c\_AE3  
  
  
# Utilities:  
  
u\_F

## [1] 0.85

Minimum\_u\_F <- 0.68  
Maximum\_u\_F <- 1.00  
  
  
u\_P

## [1] 0.65

Minimum\_u\_P <- 0.52  
Maximum\_u\_P <- 0.78   
  
  
u\_D

## [1] 0

Minimum\_u\_D <- u\_D - 0.20\*u\_D  
Maximum\_u\_D <- u\_D + 0.20\*u\_D   
  
  
AE1\_DisUtil

## [1] 0.45

Minimum\_AE1\_DisUtil <- AE1\_DisUtil - 0.20\*AE1\_DisUtil  
Maximum\_AE1\_DisUtil <- AE1\_DisUtil + 0.20\*AE1\_DisUtil   
  
  
AE2\_DisUtil

## [1] 0.19

Minimum\_AE2\_DisUtil <- AE2\_DisUtil - 0.20\*AE2\_DisUtil  
Maximum\_AE2\_DisUtil <- AE2\_DisUtil + 0.20\*AE2\_DisUtil   
  
  
AE3\_DisUtil

## [1] 0.36

Minimum\_AE3\_DisUtil <- AE3\_DisUtil - 0.20\*AE3\_DisUtil  
Maximum\_AE3\_DisUtil <- AE3\_DisUtil + 0.20\*AE3\_DisUtil   
  
  
   
   
# Discount factor  
# Cost Discount Factor  
# Utility Discount Factor  
# I divided these by 365 earlier in the R markdown document, so no need to do that again here:  
  
d\_e

## [1] 0.0001096

Minimum\_d\_e <- 0  
Maximum\_d\_e <- 0.08/365  
  
  
  
d\_c

## [1] 0.0001096

Minimum\_d\_c <- 0  
Maximum\_d\_c <- 0.08/365  
  
#   
# df\_params\_OWSA <- data.frame(  
# pars = c("HR\_FP\_Exp", "HR\_FP\_SoC", "p\_PD", "p\_FD\_SoC", "p\_FD\_Exp", "p\_FA1\_SoC", "p\_A1F\_SoC", "p\_FA2\_SoC", "p\_A2F\_SoC", "p\_FA3\_SoC", "p\_A3F\_SoC", "p\_A1D\_SoC", "p\_A2D\_SoC", "p\_A3D\_SoC"), # names of the parameters to be changed  
# min = c(Minimum\_HR\_FP\_Exp, Minimum\_HR\_FP\_SoC, Minimum\_p\_PD, Minimum\_p\_FD\_SoC, Minimum\_p\_FD\_Exp, Minimum\_p\_FA1\_SoC, Minimum\_p\_A1F\_SoC, Minimum\_p\_FA2\_SoC, Minimum\_p\_A2F\_SoC, Minimum\_p\_FA3\_SoC, Minimum\_p\_A3F\_SoC, Minimum\_p\_A1D\_SoC, Minimum\_p\_A2D\_SoC, Minimum\_p\_A3D\_SoC), # min parameter values  
# max = c(Maximum\_HR\_FP\_Exp, Maximum\_HR\_FP\_SoC, Maximum\_p\_PD, Maximum\_p\_FD\_SoC, Maximum\_p\_FD\_Exp, Maximum\_p\_FA1\_SoC, Maximum\_p\_A1F\_SoC, Maximum\_p\_FA2\_SoC, Maximum\_p\_A2F\_SoC, Maximum\_p\_FA3\_SoC, Maximum\_p\_A3F\_SoC, Maximum\_p\_A1D\_SoC, Maximum\_p\_A2D\_SoC, Maximum\_p\_A3D\_SoC) # max parameter values  
# )  
  
  
# I am concerned that the min or max may go above 1 or below 0 in cases where parameter values should be bounded at 1 or 0, therefore, in such cases I say, replace the minimum I created with 0 or the maximum I created with 1, if the minimum is below 0 or the maximum is above 1:  
  
  
HR\_FP\_Exp

## [1] 0.68

Minimum\_HR\_FP\_Exp<- replace(Minimum\_HR\_FP\_Exp, Minimum\_HR\_FP\_Exp<0, 0)  
Maximum\_HR\_FP\_Exp<- replace(Maximum\_HR\_FP\_Exp, Maximum\_HR\_FP\_Exp>1, 1)  
  
HR\_FP\_SoC

## [1] 1

Minimum\_HR\_FP\_SoC<- replace(Minimum\_HR\_FP\_SoC, Minimum\_HR\_FP\_SoC<0, 0)  
Maximum\_HR\_FP\_SoC<- replace(Maximum\_HR\_FP\_SoC, Maximum\_HR\_FP\_SoC>1, 1)  
  
HR\_PD\_SoC

## [1] 1

Minimum\_HR\_PD\_SoC<- replace(Minimum\_HR\_PD\_SoC, Minimum\_HR\_PD\_SoC<0, 0)  
Maximum\_HR\_PD\_SoC<- replace(Maximum\_HR\_PD\_SoC, Maximum\_HR\_PD\_SoC>1, 1)  
  
HR\_PD\_Exp

## [1] 0.65

Minimum\_HR\_PD\_Exp<- replace(Minimum\_HR\_PD\_Exp, Minimum\_HR\_PD\_Exp<0, 0)  
Maximum\_HR\_PD\_Exp<- replace(Maximum\_HR\_PD\_Exp, Maximum\_HR\_PD\_Exp>1, 1)  
  
P\_OSD\_SoC

## [1] 0.17

Minimum\_P\_OSD\_SoC<- replace(Minimum\_P\_OSD\_SoC, Minimum\_P\_OSD\_SoC<0, 0)  
Maximum\_P\_OSD\_SoC<- replace(Maximum\_P\_OSD\_SoC, Maximum\_P\_OSD\_SoC>1, 1)  
  
P\_OSD\_Exp

## [1] 0.17

Minimum\_P\_OSD\_Exp<- replace(Minimum\_P\_OSD\_Exp, Minimum\_P\_OSD\_Exp<0, 0)  
Maximum\_P\_OSD\_Exp<- replace(Maximum\_P\_OSD\_Exp, Maximum\_P\_OSD\_Exp>1, 1)  
  
p\_FA1\_STD

## [1] 0.04

Minimum\_p\_FA1\_STD<- replace(Minimum\_p\_FA1\_STD, Minimum\_p\_FA1\_STD<0, 0)  
Maximum\_p\_FA1\_STD<- replace(Maximum\_p\_FA1\_STD, Maximum\_p\_FA1\_STD>1, 1)  
  
p\_FA2\_STD

## [1] 0.31

Minimum\_p\_FA2\_STD<- replace(Minimum\_p\_FA2\_STD, Minimum\_p\_FA2\_STD<0, 0)  
Maximum\_p\_FA2\_STD<- replace(Maximum\_p\_FA2\_STD, Maximum\_p\_FA2\_STD>1, 1)  
  
p\_FA3\_STD

## [1] 0.31

Minimum\_p\_FA3\_STD<- replace(Minimum\_p\_FA3\_STD, Minimum\_p\_FA3\_STD<0, 0)  
Maximum\_p\_FA3\_STD<- replace(Maximum\_p\_FA3\_STD, Maximum\_p\_FA3\_STD>1, 1)  
  
p\_FA1\_EXPR

## [1] 0.07

Minimum\_p\_FA1\_EXPR<- replace(Minimum\_p\_FA1\_EXPR, Minimum\_p\_FA1\_EXPR<0, 0)  
Maximum\_p\_FA1\_EXPR<- replace(Maximum\_p\_FA1\_EXPR, Maximum\_p\_FA1\_EXPR>1, 1)  
  
p\_FA2\_EXPR

## [1] 0.11

Minimum\_p\_FA2\_EXPR<- replace(Minimum\_p\_FA2\_EXPR, Minimum\_p\_FA2\_EXPR<0, 0)  
Maximum\_p\_FA2\_EXPR<- replace(Maximum\_p\_FA2\_EXPR, Maximum\_p\_FA2\_EXPR>1, 1)  
  
p\_FA3\_EXPR

## [1] 0.07

Minimum\_p\_FA3\_EXPR<- replace(Minimum\_p\_FA3\_EXPR, Minimum\_p\_FA3\_EXPR<0, 0)  
Maximum\_p\_FA3\_EXPR<- replace(Maximum\_p\_FA3\_EXPR, Maximum\_p\_FA3\_EXPR>1, 1)  
  
u\_F

## [1] 0.85

Minimum\_u\_F<- replace(Minimum\_u\_F, Minimum\_u\_F<0, 0)  
Maximum\_u\_F<- replace(Maximum\_u\_F, Maximum\_u\_F>1, 1)  
  
u\_P

## [1] 0.65

Minimum\_u\_P<- replace(Minimum\_u\_P, Minimum\_u\_P<0, 0)  
Maximum\_u\_P<- replace(Maximum\_u\_P, Maximum\_u\_P>1, 1)  
  
AE1\_DisUtil

## [1] 0.45

Minimum\_AE1\_DisUtil<- replace(Minimum\_AE1\_DisUtil, Minimum\_AE1\_DisUtil<0, 0)  
Maximum\_AE1\_DisUtil<- replace(Maximum\_AE1\_DisUtil, Maximum\_AE1\_DisUtil>1, 1)  
  
AE2\_DisUtil

## [1] 0.19

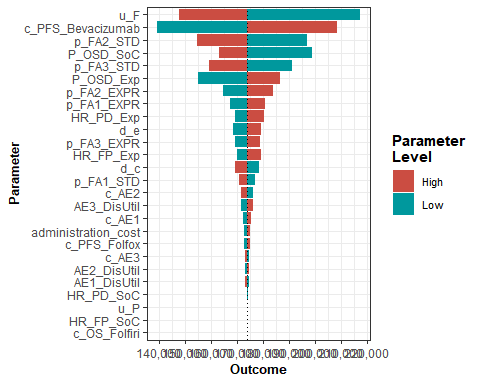
Minimum\_AE2\_DisUtil<- replace(Minimum\_AE2\_DisUtil, Minimum\_AE2\_DisUtil<0, 0)  
Maximum\_AE2\_DisUtil<- replace(Maximum\_AE2\_DisUtil, Maximum\_AE2\_DisUtil>1, 1)  
  
AE3\_DisUtil

## [1] 0.36

Minimum\_AE3\_DisUtil<- replace(Minimum\_AE3\_DisUtil, Minimum\_AE3\_DisUtil<0, 0)  
Maximum\_AE3\_DisUtil<- replace(Maximum\_AE3\_DisUtil, Maximum\_AE3\_DisUtil>1, 1)  
  
  
  
# A one-way sensitivity analysis (OWSA) can be defined by specifying the names of the parameters that are to be incuded and their minimum and maximum values.  
  
  
# We create a dataframe containing all parameters we want to do the sensitivity analysis on, and the min and max values of the parameters of interest   
# "min" and "max" are the mininum and maximum values of the parameters of interest.  
  
  
# options(scipen = 999) # disabling scientific notation in R  
  
df\_params\_OWSA <- data.frame(  
 pars = c("HR\_FP\_Exp", "HR\_FP\_SoC", "HR\_PD\_SoC", "HR\_PD\_Exp", "P\_OSD\_SoC", "P\_OSD\_Exp", "p\_FA1\_STD", "p\_FA2\_STD", "p\_FA3\_STD", "p\_FA1\_EXPR", "p\_FA2\_EXPR", "p\_FA3\_EXPR", "administration\_cost", "c\_PFS\_Folfox", "c\_PFS\_Bevacizumab", "c\_OS\_Folfiri", "c\_AE1", "c\_AE2", "c\_AE3", "d\_e", "d\_c", "u\_F", "u\_P", "AE1\_DisUtil", "AE2\_DisUtil", "AE3\_DisUtil"), # names of the parameters to be changed  
 min = c(Minimum\_HR\_FP\_Exp, Minimum\_HR\_FP\_SoC, Minimum\_HR\_PD\_SoC, Minimum\_HR\_PD\_Exp, Minimum\_P\_OSD\_SoC, Minimum\_P\_OSD\_Exp, Minimum\_p\_FA1\_STD, Minimum\_p\_FA2\_STD, Minimum\_p\_FA3\_STD, Minimum\_p\_FA1\_EXPR, Minimum\_p\_FA2\_EXPR, Minimum\_p\_FA3\_EXPR, Minimum\_administration\_cost, Minimum\_c\_PFS\_Folfox, Minimum\_c\_PFS\_Bevacizumab, Minimum\_c\_OS\_Folfiri, Minimum\_c\_AE1, Minimum\_c\_AE2, Minimum\_c\_AE3, Minimum\_d\_e, Minimum\_d\_c, Minimum\_u\_F, Minimum\_u\_P, Minimum\_AE1\_DisUtil, Minimum\_AE2\_DisUtil, Minimum\_AE3\_DisUtil), # min parameter values  
 max = c(Maximum\_HR\_FP\_Exp, Maximum\_HR\_FP\_SoC, Maximum\_HR\_PD\_SoC, Maximum\_HR\_PD\_Exp, Maximum\_P\_OSD\_SoC, Maximum\_P\_OSD\_Exp, Maximum\_p\_FA1\_STD, Maximum\_p\_FA2\_STD, Maximum\_p\_FA3\_STD, Maximum\_p\_FA1\_EXPR, Maximum\_p\_FA2\_EXPR, Maximum\_p\_FA3\_EXPR, Maximum\_administration\_cost, Maximum\_c\_PFS\_Folfox, Maximum\_c\_PFS\_Bevacizumab, Maximum\_c\_OS\_Folfiri, Maximum\_c\_AE1, Maximum\_c\_AE2, Maximum\_c\_AE3, Maximum\_d\_e, Maximum\_d\_c, Maximum\_u\_F, Maximum\_u\_P, Maximum\_AE1\_DisUtil, Maximum\_AE2\_DisUtil, Maximum\_AE3\_DisUtil) # max parameter values  
)  
  
   
  
  
  
  
  
  
  
  
# I made sure the names of the parameters to be varied and their mins and maxs are in the same order in all the brackets above in order to make sure that the min and max being applied are the min and the max of the parameter I want to consider a min and a max for.  
  
  
  
# The OWSA is performed using the run\_owsa\_det function  
  
  
# This function runs a deterministic one-way sensitivity analysis (OWSA) on a given function that produces outcomes. rdrr.io/github/DARTH-git/dampack/src/R/run\_dsa.R  
  
DSAICER <- run\_owsa\_det(  
  
# run\_owsa\_det: https://rdrr.io/github/DARTH-git/dampack/man/run\_owsa\_det.html  
  
 # We need to make sure we consistently use "DSAICER" throughout, or else the function will present with an error saying "DSAICER" not found.   
   
# Arguments:  
   
 params\_range = df\_params\_OWSA, # dataframe with parameters for OWSA  
  
# params\_range   
# data.frame with 3 columns of parameters for OWSA in the following order: "pars", "min", and "max".  
# The number of samples from this range is determined by nsamp.   
# "pars" are the parameters of interest and must be a subset of the parameters from params\_basecase.  
  
  
# Details  
# params\_range  
   
# "pars" are the names of the input parameters of interest. These are the parameters that will be varied in the deterministic sensitivity analysis. variables in "pars" column must be a subset of variables in params\_basecase  
   
  
   
 params\_basecase = l\_params\_all, # list with all parameters  
  
# params\_basecase   
  
# a named list of basecase values for input parameters needed by FUN, the user-defined function. So, I guess it takes the values that the parameters are equal to in l\_params\_all as the base case, so if cost is generated equal to 1,000 it'll take that as the base case, and then take the min and the max around this from the data.frame we created above.  
  
# To conduct the one-way DSA, we then call the function run\_owsa\_det with my\_owsa\_params\_range, specifying the parameters to be varied and over which ranges, and my\_params\_basecase as the fixed parameter values to be used in the model when a parameter is not being explicitly varied in the one-way sensitivity analysis. https://cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
  
 nsamp = 100, # number of parameter values  
  
# nsamp   
  
# number of sets of parameter values to be generated. If NULL, 100 parameter values are used -> I think Eva Enns said these are automatically evenly spaced out values of the parameters.  
  
# Additional inputs are the number of equally-spaced samples (nsamp) to be used between the specified minimum and maximum of each range, the user-defined function (FUN) to be called to generate model outcomes for each strategy, the vector of outcomes to be stored (must be outcomes generated by the data frame output of the function passed in FUN), and the vector of strategy names to be evaluated.  
  
# cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
  
 FUN = oncologySemiMarkov, # function to compute outputs  
  
# FUN   
# function that takes the basecase in params\_basecase and runs the analysis in the function... to produce the outcome of interest. The FUN must return a dataframe where the first column are the strategy names and the rest of the columns must be outcomes.  
#   
  
 outcomes = c("DSAICER"), # output to do the OWSA on  
  
# string vector with the outcomes of interest from FUN produced by nsamp  
# This basically tells run\_owsa\_det what the name of the outcome of interest from the function we fed it is. Here our function previously had NMB, i.e., the net monetary benefit.  
  
# outcomes = c("NMB"), # output to do the OWSA on  
  
# outcomes   
  
 strategies = v\_names\_strats, # names of the strategies  
  
# strategies  
# Set it equal to a vector of strategy names. The default NULL will use strategy names in FUN ( strategies = NULL,)  
# Here that's "Standard of Care" and "Experimental Treatment".  
  
 progress = TRUE,  
  
# progress   
# TRUE or FALSE for whether or not function progress should be displayed in console, i.e., like 75% complete, 100%, etc.,  
  
# The input progress = TRUE allows the user to see a progress bar as the DSA is conducted. When many parameters are being varied, nsamp is large, and/or the user-defined function is computationally burdensome, the DSA may take a noticeable amount of time to compute and the progress display is recommended.  
# cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
  
  
 n\_wtp = 45000 # extra argument to pass to FUN to specify the willingness to pay  
)

## | | | 0% | | | 1% | |= | 1% | |= | 2% | |== | 2% | |== | 3% | |== | 4% | |=== | 4% | | | 0% | |=== | 4% | |=== | 5% | |==== | 5% | |==== | 6% | |===== | 6% | |===== | 7% | |===== | 8% | | | 0% | |===== | 8% | |====== | 8% | |====== | 9% | |======= | 9% | |======= | 10% | |======= | 11% | |======== | 11% | |======== | 12% | | | 0% | |======== | 12% | |========= | 12% | |========= | 13% | |========= | 14% | |========== | 14% | |========== | 15% | |=========== | 15% | | | 0% | |=========== | 15% | |=========== | 16% | |============ | 16% | |============ | 17% | |============ | 18% | |============= | 18% | |============= | 19% | | | 0% | |============= | 19% | |============== | 19% | |============== | 20% | |============== | 21% | |=============== | 21% | |=============== | 22% | |================ | 22% | |================ | 23% | | | 0% | |================ | 23% | |================ | 24% | |================= | 24% | |================= | 25% | |================== | 25% | |================== | 26% | |=================== | 26% | |=================== | 27% | | | 0% | |=================== | 27% | |=================== | 28% | |==================== | 28% | |==================== | 29% | |===================== | 29% | |===================== | 30% | |===================== | 31% | |====================== | 31% | | | 0% | |====================== | 31% | |====================== | 32% | |======================= | 32% | |======================= | 33% | |======================= | 34% | |======================== | 34% | |======================== | 35% | | | 0% | |======================== | 35% | |========================= | 35% | |========================= | 36% | |========================== | 36% | |========================== | 37% | |========================== | 38% | |=========================== | 38% | | | 0% | |=========================== | 38% | |=========================== | 39% | |============================ | 39% | |============================ | 40% | |============================ | 41% | |============================= | 41% | |============================= | 42% | |============================== | 42% | | | 0% | |============================== | 42% | |============================== | 43% | |============================== | 44% | |=============================== | 44% | |=============================== | 45% | |================================ | 45% | |================================ | 46% | | | 0% | |================================ | 46% | |================================= | 46% | |================================= | 47% | |================================= | 48% | |================================== | 48% | |================================== | 49% | |=================================== | 49% | |=================================== | 50% | | | 0% | |=================================== | 50% | |=================================== | 51% | |==================================== | 51% | |==================================== | 52% | |===================================== | 52% | |===================================== | 53% | |===================================== | 54% | |====================================== | 54% | | | 0% | |====================================== | 54% | |====================================== | 55% | |======================================= | 55% | |======================================= | 56% | |======================================== | 56% | |======================================== | 57% | |======================================== | 58% | | | 0% | |======================================== | 58% | |========================================= | 58% | |========================================= | 59% | |========================================== | 59% | |========================================== | 60% | |========================================== | 61% | |=========================================== | 61% | |=========================================== | 62% | | | 0% | |=========================================== | 62% | |============================================ | 62% | |============================================ | 63% | |============================================ | 64% | |============================================= | 64% | |============================================= | 65% | |============================================== | 65% | | | 0% | |============================================== | 65% | |============================================== | 66% | |=============================================== | 66% | |=============================================== | 67% | |=============================================== | 68% | |================================================ | 68% | |================================================ | 69% | | | 0% | |================================================ | 69% | |================================================= | 69% | |================================================= | 70% | |================================================= | 71% | |================================================== | 71% | |================================================== | 72% | |=================================================== | 72% | |=================================================== | 73% | | | 0% | |=================================================== | 73% | |=================================================== | 74% | |==================================================== | 74% | |==================================================== | 75% | |===================================================== | 75% | |===================================================== | 76% | |====================================================== | 76% | |====================================================== | 77% | | | 0% | |====================================================== | 77% | |====================================================== | 78% | |======================================================= | 78% | |======================================================= | 79% | |======================================================== | 79% | |======================================================== | 80% | |======================================================== | 81% | |========================================================= | 81% | | | 0% | |========================================================= | 81% | |========================================================= | 82% | |========================================================== | 82% | |========================================================== | 83% | |========================================================== | 84% | |=========================================================== | 84% | |=========================================================== | 85% | | | 0% | |=========================================================== | 85% | |============================================================ | 85% | |============================================================ | 86% | |============================================================= | 86% | |============================================================= | 87% | |============================================================= | 88% | |============================================================== | 88% | | | 0% | |============================================================== | 88% | |============================================================== | 89% | |=============================================================== | 89% | |=============================================================== | 90% | |=============================================================== | 91% | |================================================================ | 91% | |================================================================ | 92% | |================================================================= | 92% | | | 0% | |================================================================= | 92% | |================================================================= | 93% | |================================================================= | 94% | |================================================================== | 94% | |================================================================== | 95% | |=================================================================== | 95% | |=================================================================== | 96% | | | 0% | |=================================================================== | 96% | |==================================================================== | 96% | |==================================================================== | 97% | |==================================================================== | 98% | |===================================================================== | 98% | |===================================================================== | 99% | |======================================================================| 99% | |======================================================================| 100%

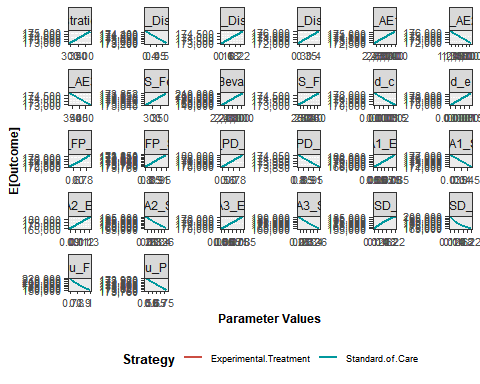
# Value  
# A list containing dataframes with the results of the sensitivity analyses. The list will contain a dataframe for each outcome specified. List elements can be visualized with plot.owsa, owsa\_opt\_strat and owsa\_tornado from dampack  
  
# Basically, run\_owsa\_det creates the above.  
  
# Also, each owsa object returned by run\_owsa\_det is a data frame with four columns: parameter, strategy, param\_val, and outcome\_val. For each row, param\_val is the value used for the parameter listed in parameter and outcome\_val is the value of the specified outcome for the strategy listed in strategy.   
  
# So, OWSA\_NMB shows a row is created for each sampling of the parameter value from min to max, so 100 rows per parameter (because nsamp = 100,), and the outcome value associated with a parameter value of this size is displayed under outcome\_val.  
  
# You can see how this works by putting the below into the console.  
  
# OWSA\_NMB  
  
# Or just DSAICER for DSAICER.  
  
# Resources on this available here:  
  
  
# https://rdrr.io/github/DARTH-git/dampack/man/run\_owsa\_det.html (also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\sensitivity analysis help files\rdrr-io-github-DARTH-git-dampack-man-run\_owsa\_det-html.pdf)  
  
  
# This link is pretty good for showing that the OWSA can be run across a variety of outcomes outcomes = c("Cost", "QALY", "LY", "NMB"), and then you can pick which of the OWSA outcomes you'd like to focus on and build tornado diagrams, etc., from that:  
  
# "Because we have defined multiple parameters in my\_owsa\_params\_range, we have instructed run\_owsa\_det to execute a series of separate one-way sensitivity analyses and compile the results into a single owsa object for each requested outcome. When only one outcome is specified run\_owsa\_det returns a owsa data frame. When more than one outcome is specified, run\_owsa\_det returns a list containing one owsa data frame for each outcome. To access the owsa object corresponding to a given outcome, one can select the list item with the name “owsa\_”. For example, the owsa object associated with the NMB outcome can be accessed as l\_owsa\_det$owsa\_NMB."  
  
# https://cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html   
  
owsa\_tornado(owsa = DSAICER, txtsize = 11)



# Plotting the outcomes of the OWSA in a tornado plot  
# - note that other plots can also be generated using the plot() and owsa\_opt\_strat() functions  
  
  
  
# owsa\_tornado(DSAICER,  
#   
# min\_rel\_diff = 0.05)  
  
# For owsa objects that contain many parameters that have minimal effect on the parameter of interest, you may want to consider producing a plot that highlights only the most influential parameters. Using the min\_rel\_diff argument, you can instruct owsa\_tornado to exclude all parameters that fail to produce a relative change in the outcome below a specific fraction.  
  
  
# Maybe use the below to change the formatting of the above:  
# TornadoPlot(main\_title = "Tornado Plot", Parms = paramNames, Outcomes = m.tor,   
# outcomeName = "Incremental Cost-Effectiveness Ratio (ICER)",   
# xlab = "ICER",   
# ylab = "Parameters",   
# col1="#3182bd", col2="#6baed6")

## 08.3.1 Plot OWSA

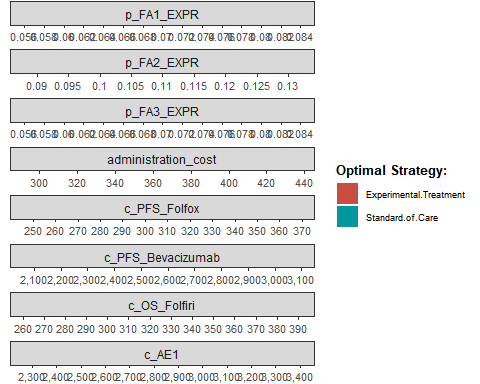
plot(DSAICER, txtsize = 10, n\_x\_ticks = 4,   
 facet\_scales = "free") +  
 theme(legend.position = "bottom")



# I can add the following to the above:   
  
# + ggtitle("Expected Value of Perfect Information")  
  
# To include a title on this plot, per: https://mran.microsoft.com/snapshot/2021-03-21/web/packages/dampack/dampack.pdf  
  
#txtsize - base text size  
  
# n\_y\_ticks - number of axis ticks  
# n\_x\_ticks - number of axis ticks  
  
# Function for determining number of ticks on axis of ggplot2 plots. https://www.quantargo.com/help/r/latest/packages/dampack/1.0.1/number\_ticks  
  
  
# [SO, IN THE DIAGRAM THE PARAMETER WE ARE CONSIDERING IS AT THE TOP OF THE DIAGRAM, THE NET MONETARY BENEFIT IS ON THE LEFT OF THE DIAGRAM AND YOU'LL SEE THAT THE LEFT EDGE OF THE LINE IS THE MINIMUM VALUE FOR THE PARAMETER WE ARE INTERESTED IN, THE RIGHT EDGE OF THE LINE IS THE MAXIMUM VALUE FOR THE PARAMETER WE ARE INTERESTED IN, AND SOMEWHERE IN THE MIDDLE IS THE BASECASE WE ASSIGNED THIS PARAMETER. A GOOD EXAMPLE OF THIS IS c\_P, COST OF PROGRESSION, WHERE THE BASECASE WAS 1,000 EURO, WHICH WE SEE IN THE MIDDLE, THE MINIMUM IS 800 EURO WHICH WE SEE ON THE FAR LEFT OF THE LINE, AND THE MAXIMUM IS 1,200 WHICH WE SEE ON THE FAR RIGHT OF THE LINE, AND WE SEE THAT AS WE MOVE FROM THE MINUMUM, THE 800 EURO ON THE FAR LEFT OF THE LINE, TO THE MAXIMUM, THE 1200 EURO ON THE FAR RIGHT OF THE LINE (I.E., AS THE COST OF BEING IN THE PROGRESSED STATE INCREASES) THE NET MONETARY BENEFIT OF BOTH TREATMENT OPTIONS FALL, WITH THE NMB OF THE EXPERIMENTAL TREATMENT OPTION FALLING FROM 235,000 TO 230,000].  
  
  
  
# Explained here: https://cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
  
  
# and here: https://cran.r-project.org/web/packages/dampack/vignettes/psa\_analysis.html  
  
  
# Because I'm using ICERs, there's only one value, whereas for NMB, there's a value under each treatment strategy, but ICERs are ratios comparing the costs of one treatment strategy to another, and the outcomes of one treatment strategy to another. So, where when I was using NMB as my outcome I would have a curve for "SoC" and "Experimental Treatment", because each would get an NMB, now I only have the one curve, to show how the ICER value changes. So, before I could have how the net monetary benefit changed under each treatment strategy as the cost of treatment A went up, or the rate of changing from healthy to dead went up, per the "# Plot outcome of each strategy over each parameter range" on: https://cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html, now I only have the ICER, so I see how the ICER changes as the cost of parameters, etc., change.  
  
# Because only one ICER value is created, but it's created for both SoC and Exp, I imagine that the exp line and SoC line are created, but sit on top of eachother in the diagram below:

## 08.3.2 Optimal strategy with OWSA

# There are too many parameters to make things legible, so instead you should create a tornado plot that highlights only the most influential parameters. Using the min\_rel\_diff argument, you can instruct owsa\_tornado to exclude all parameters that fail to produce a relative change in the outcome below a specific fraction.  
  
# owsa\_tornado(OWSA\_NMB,  
#   
# min\_rel\_diff = 0.05)  
  
# Following this, create a vector of the parameters that appear in this tornado plot below and then create the optimal strategy with OWSA per these parameters.  
  
# v\_owsa\_opt\_strat <- c("HR\_FP\_Exp", "HR\_FP\_SoC", "p\_PD", "p\_FD\_SoC", "p\_FD\_Exp", "p\_FA1\_SoC", "p\_A1F\_SoC", "p\_A1D\_SoC", "p\_FA2\_SoC", "p\_A2F\_SoC", "p\_A2D\_SoC", "p\_FA3\_SoC", "p\_A3F\_SoC", "p\_A3D\_SoC", "c\_F\_SoC", "c\_F\_Exp", "c\_P","c\_AE1", "c\_AE2", "c\_AE3", "d\_e", "d\_c", "u\_F", "u\_P", "u\_AE1", "u\_AE2", "u\_AE3")  
  
# Then include params = v\_owsa\_opt\_strat, in the below and get rid of plot\_const = FALSE  
  
owsa\_opt\_strat(owsa = DSAICER, txtsize = 10, plot\_const = FALSE)



# plot\_const = FALSE  
# plot\_const   
# whether to plot parameters that don't lead to changes in optimal strategy as they vary.  
  
# params   
# params = v\_owsa\_opt\_strat,  
# vector of parameters to plot  
  
# return   
# either return a ggplot object plot or a data frame with ranges of parameters for which each strategy is optimal. return = c("plot", "data"),  
  
# Basically, the data frame will show you the ranges for each parameter that each strategy is optimal.You can see below that for p\_FD\_Exp the experimental treatment is the optimal strategy from 0.04 (the minimum) through to 0.055 (0.05575757575757575), while standard of care is the optimal strategy from 0.059 (0.05595959595959596) through to 0.06 (the maximum).   
  
# p\_FD\_Exp Experimental.Treatment 4.0000000000000001e-02 5.5757575757575756e-02   
# p\_FD\_Exp Standard.of.Care 5.5959595959595959e-02 6.0000000000000005e-02   
  
# Whereas for p\_FA3\_SoC the experimental treatment strategy is optimal from it's min 0.016 all the way through every value in the range to it's max: 0.024   
  
# p\_FA3\_SoC Experimental.Treatment 1.6000000000000000e-02 2.4000000000000000e-02   
  
# The data frame is a bit of a pain though because it returns things in scientific notation, but according to the below:  
  
   
# Scientific Notation  
# E+01 means moving the decimal point one digit to the right, E+00 means leaving the decimal point where it is, and E–01 means moving the decimal point one digit to the left. Example: 1.00E+01 is 10, 1.33E+00 stays at 1.33, and 1.33E–01 becomes 0.133.  
# https://neo.ne.gov/programs/stats/inf/79.htm#:~:text=Scientific%20Notation,-The%20scientific%20notation&text=E%2B01%20means%20moving%20the,1.33E%E2%80%9301%20becomes%200.133.  
  
# There's also a good calculator to do this for you here: https://www.free-online-calculator-use.com/scientific-notation-converter.html#  
  
# https://rdrr.io/github/DARTH-git/dampack/man/owsa\_opt\_strat.html  
  
  
# Like owsa\_tornado(), the return argument in owsa\_opt\_strat() allows the user to access a tidy data.frame that contains the exact values used to produce the plot.  
  
# owsa\_opt\_strat(o,   
# return = "data")  
  
# https://cran.r-project.org/web/packages/dampack/vignettes/psa\_analysis.html  
  
  
  
# If return == "plot", a ggplot2 optimal strategy plot derived from the owsa object, or if return == "data", a data.frame containing all data contained in the plot. The plot allows us to see how the strategy that maximizes the expectation of the outcome of interest changes as a function of each parameter of interest.  
  
# Visualize optimal strategy (max NMB) over each parameter range  
# owsa\_opt\_strat(my\_owsa\_NMB)  
  
# facet\_ncol   
# Number of columns in plot facet.  
  
# facet\_nrow   
# number of rows in plot facet.  
  
# txtsize  
# base text size  
  
# facet\_lab\_txtsize   
# text size for plot facet labels (I think this allows me change the size of the text on the plots and the size of the numbers, ledgend, etc., independently).  
  
# Explained here:   
  
# https://cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
  
# https://cran.r-project.org/web/packages/dampack/vignettes/psa\_analysis.html  
  
# https://rdrr.io/github/DARTH-git/dampack/man/owsa\_opt\_strat.html  
  
# https://www.quantargo.com/help/r/latest/packages/dampack/1.0.1/owsa\_opt\_strat  
  
# Again, this is something that doesnt work too well with ICERs, with a NMB it is easy to see what the optimal strategy is, with am ICER it's not so easy unless you are telling your code what the willingness to pay threshold is, and it's doing the ICER calculation and then comparing this to the WTP threshold to see which approach is most cost-effective. But I think this code is really set up for NMB and telling you what parameter values give you higher NMB values and which treatment strategies these NMB's belong to.

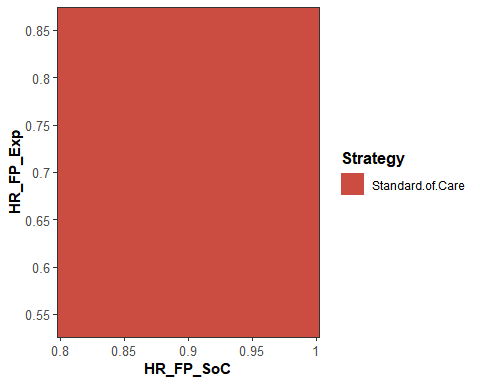
## 

## 08.4 Two-way sensitivity analysis (TWSA)

# To conduct the two-way DSA, we call the function run\_twsa\_det with my\_twsa\_params\_range. The general format of the function arguments for run\_twsa\_det are the same as those for run\_owsa\_det. In run\_twsa\_det, equally spaced sequences of length nsamp are created for the two parameters based on the inputs provided in the params\_range argument. These two sequences of parameter values define an nsamp by nsamp grid over which FUN is applied to produce outcomes for every combination of the two parameters. https://cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
  
  
## 4.2 Defining and performing a two-way sensitivity analysis ----  
  
# I can also preform a TWSA on a a probabilistic sensitivity analysis (make\_psa\_obj) or a deterministic sensitivity analysis object (run\_owsa\_det) per: https://rdrr.io/github/DARTH-git/dampack/man/twsa.html  
  
# A lot of this code should be reflective of the earlier code of the one-way sensitivity analysis, so if I'm confused as to what one thing is doing or another I can look at the piece of code that is confusing me here, and scroll back up to my explanation of that code in OWSA.  
  
# Run deterministic two-way sensitivity analysis (TWSA) https://rdrr.io/github/DARTH-git/dampack/src/R/run\_dsa.R  
  
# To perform a two-way sensitivity analysis (TWSA), a similar data.frame with model parameters is required  
  
# dataframe containing all parameters, their basecase values, and the min and   
# max values of the parameters of interest  
  
df\_params\_TWSA <- data.frame(pars = c("HR\_FP\_SoC", "HR\_FP\_Exp"),  
 min = c(Minimum\_HR\_FP\_SoC, Minimum\_HR\_FP\_Exp), # min parameter values  
 max = c(Maximum\_HR\_FP\_SoC, Maximum\_HR\_FP\_Exp) # max parameter values  
)  
  
# We could have chosen any of the below from our OWSA data.frame to use as our parameters:  
# df\_params\_OWSA <- data.frame(  
# pars = c("HR\_FP\_Exp", "HR\_FP\_SoC", "p\_PD", "p\_FD\_SoC", "p\_FD\_Exp", "p\_FA1\_SoC", "p\_A1F\_SoC", "p\_A1D\_SoC", "p\_FA2\_SoC", "p\_A2F\_SoC", "p\_A2D\_SoC", "p\_FA3\_SoC", "p\_A3F\_SoC", "p\_A3D\_SoC", "c\_F\_SoC", "c\_F\_Exp", "c\_P","c\_AE1", "c\_AE2", "c\_AE3", "d\_e", "d\_c", "u\_F", "u\_P", "u\_AE1", "u\_AE2", "u\_AE3"), # names of the parameters to be changed  
# min = c(Minimum\_HR\_FP\_Exp, Minimum\_HR\_FP\_SoC, Minimum\_p\_PD, Minimum\_p\_FD\_SoC, Minimum\_p\_FD\_Exp, Minimum\_p\_FA1\_SoC, Minimum\_p\_A1F\_SoC, Minimum\_p\_A1D\_SoC, Minimum\_p\_FA2\_SoC, Minimum\_p\_A2F\_SoC, Minimum\_p\_A2D\_SoC, Minimum\_p\_FA3\_SoC, Minimum\_p\_A3F\_SoC, Minimum\_p\_A3D\_SoC, Minimum\_c\_F\_SoC, Minimum\_c\_F\_Exp, Minimum\_c\_P, Minimum\_c\_AE1, Minimum\_c\_AE2, Minimum\_c\_AE3, Minimum\_d\_e, Minimum\_d\_c, Minimum\_u\_F, Minimum\_u\_P, Minimum\_u\_AE1, Minimum\_u\_AE2, Minimum\_u\_AE3), # min parameter values  
# max = c(Maximum\_HR\_FP\_Exp, Maximum\_HR\_FP\_SoC, Maximum\_p\_PD, Maximum\_p\_FD\_SoC, Maximum\_p\_FD\_Exp, Maximum\_p\_FA1\_SoC, Maximum\_p\_A1F\_SoC, Maximum\_p\_A1D\_SoC, Maximum\_p\_FA2\_SoC, Maximum\_p\_A2F\_SoC, Maximum\_p\_A2D\_SoC, Maximum\_p\_FA3\_SoC, Maximum\_p\_A3F\_SoC, Maximum\_p\_A3D\_SoC, Maximum\_c\_F\_SoC, Maximum\_c\_F\_Exp, Maximum\_c\_P, Maximum\_c\_AE1, Maximum\_c\_AE2, Maximum\_c\_AE3, Maximum\_d\_e, Maximum\_d\_c, Maximum\_u\_F, Maximum\_u\_P, Maximum\_u\_AE1, Maximum\_u\_AE2, Maximum\_u\_AE3) # max parameter values  
# )  
#   
  
# It's a pain, but we have to only enter 2 parameters of interest at a time into our model per the error built into the source code for run\_twsa\_det:  
  
# "two-way sensitivity analysis only allows for and requires 2 different paramters of interest at a time" https://rdrr.io/github/DARTH-git/dampack/src/R/run\_dsa.R  
  
# And:  
  
# The structure of the function is very similar to run\_owsa\_det(). The primary difference is the function can only take two parameters at a time in the params\_range. https://syzoekao.github.io/CEAutil/  
  
  
# The TWSA is performed using the run\_twsa\_det function  
  
  
TWSA\_DSAICER <- run\_twsa\_det(params\_range = df\_params\_TWSA, # dataframe with parameters for TWSA  
 params\_basecase = l\_params\_all, # list with all parameters, the "pars" chosen in the data.frame to be analysed here must be a subset of these.  
   
 nsamp = 40, # number of parameter values. If NULL, 40 parameter values are used  
  
 # number of parameter values  
  
# nsamp   
  
# number of sets of parameter values to be generated. If NULL, 40 parameter values are used -> I think Eva Enns said these are automatically evenly spaced out values of the parameters.  
  
# Additional inputs are the number of equally-spaced samples (nsamp) to be used between the specified minimum and maximum of each range, the user-defined function (FUN) to be called to generate model outcomes for each strategy, the vector of outcomes to be stored (must be outcomes generated by the data frame output of the function passed in FUN), and the vector of strategy names to be evaluated.  
  
# cran.r-project.org/web/packages/dampack/vignettes/dsa\_generation.html  
   
   
 FUN = oncologySemiMarkov, # function to compute outputs  
  
# Function that takes the basecase in params\_all and produces the outcome of interest. The FUN must return a dataframe where the first column is the strategy names and the rest of the columns must be outcomes. Which df\_ce in the function does.  
  
# Described here:  
  
# https://rdrr.io/cran/dampack/man/run\_twsa\_det.html  
  
 outcomes = c("DSAICER"), # output to do the TWSA on  
 strategies = v\_names\_strats, # names of the strategies.The default (NULL) will use strategy names in FUN  
 progress = TRUE, #Progress bar like before  
  
 n\_wtp = 45000 # extra argument to pass to FUN to specify the willingness to pay  
)

## | | | 0% | | | 1% | |= | 1% | |= | 2% | |== | 2% | |== | 3% | |== | 4% | |=== | 4% | |=== | 5% | |==== | 5% | |==== | 6% | |===== | 6% | |===== | 7% | |===== | 8% | |====== | 8% | |====== | 9% | |======= | 9% | |======= | 10% | |======= | 11% | |======== | 11% | |======== | 12% | |========= | 12% | |========= | 13% | |========= | 14% | |========== | 14% | |========== | 15% | |=========== | 15% | |=========== | 16% | |============ | 16% | |============ | 17% | |============ | 18% | |============= | 18% | |============= | 19% | |============== | 19% | |============== | 20% | |============== | 21% | |=============== | 21% | |=============== | 22% | |================ | 22% | |================ | 23% | |================ | 24% | |================= | 24% | |================= | 25% | |================== | 25% | |================== | 26% | |=================== | 26% | |=================== | 27% | |=================== | 28% | |==================== | 28% | |==================== | 29% | |===================== | 29% | |===================== | 30% | |===================== | 31% | |====================== | 31% | |====================== | 32% | |======================= | 32% | |======================= | 33% | |======================= | 34% | |======================== | 34% | |======================== | 35% | |========================= | 35% | |========================= | 36% | |========================== | 36% | |========================== | 37% | |========================== | 38% | |=========================== | 38% | |=========================== | 39% | |============================ | 39% | |============================ | 40% | |============================ | 41% | |============================= | 41% | |============================= | 42% | |============================== | 42% | |============================== | 43% | |============================== | 44% | |=============================== | 44% | |=============================== | 45% | |================================ | 45% | |================================ | 46% | |================================= | 46% | |================================= | 47% | |================================= | 48% | |================================== | 48% | |================================== | 49% | |=================================== | 49% | |=================================== | 50% | |=================================== | 51% | |==================================== | 51% | |==================================== | 52% | |===================================== | 52% | |===================================== | 53% | |===================================== | 54% | |====================================== | 54% | |====================================== | 55% | |======================================= | 55% | |======================================= | 56% | |======================================== | 56% | |======================================== | 57% | |======================================== | 58% | |========================================= | 58% | |========================================= | 59% | |========================================== | 59% | |========================================== | 60% | |========================================== | 61% | |=========================================== | 61% | |=========================================== | 62% | |============================================ | 62% | |============================================ | 63% | |============================================ | 64% | |============================================= | 64% | |============================================= | 65% | |============================================== | 65% | |============================================== | 66% | |=============================================== | 66% | |=============================================== | 67% | |=============================================== | 68% | |================================================ | 68% | |================================================ | 69% | |================================================= | 69% | |================================================= | 70% | |================================================= | 71% | |================================================== | 71% | |================================================== | 72% | |=================================================== | 72% | |=================================================== | 73% | |=================================================== | 74% | |==================================================== | 74% | |==================================================== | 75% | |===================================================== | 75% | |===================================================== | 76% | |====================================================== | 76% | |====================================================== | 77% | |====================================================== | 78% | |======================================================= | 78% | |======================================================= | 79% | |======================================================== | 79% | |======================================================== | 80% | |======================================================== | 81% | |========================================================= | 81% | |========================================================= | 82% | |========================================================== | 82% | |========================================================== | 83% | |========================================================== | 84% | |=========================================================== | 84% | |=========================================================== | 85% | |============================================================ | 85% | |============================================================ | 86% | |============================================================= | 86% | |============================================================= | 87% | |============================================================= | 88% | |============================================================== | 88% | |============================================================== | 89% | |=============================================================== | 89% | |=============================================================== | 90% | |=============================================================== | 91% | |================================================================ | 91% | |================================================================ | 92% | |================================================================= | 92% | |================================================================= | 93% | |================================================================= | 94% | |================================================================== | 94% | |================================================================== | 95% | |=================================================================== | 95% | |=================================================================== | 96% | |==================================================================== | 96% | |==================================================================== | 97% | |==================================================================== | 98% | |===================================================================== | 98% | |===================================================================== | 99% | |======================================================================| 99% | |======================================================================| 100%

# Plot TWSA  
  
plot(TWSA\_DSAICER)



# Sometimes you see "maximise" written in people's TWSA code, why is that:  
  
# maximize   
# If TRUE, plot of strategy with maximum expected outcome (default); if FALSE, plot of strategy with minimum expected outcome  
  
# per: https://www.quantargo.com/help/r/latest/packages/dampack/1.0.1/plot.twsa  
  
  
  
# [[A 2-way uncertainty analysis will be more useful if informed by the covariance between the 2 parameters of interest or on the logical relationship between them (e.g., a 2-way uncertainty analysis might be represented by the control intervention event rate and the hazard ratio with the new treatment). <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>]]  
  
# The Briggs 2012 paper said to include 2 parameters in the TWSA that have a logical relationship,(e.g., a 2-way uncertainty analysis might be represented by the control intervention event rate and the hazard ratio with the new treatment), which are expected to have a relationship because the hazard ratio for the experimental strategy is multiplied by the rate of events under soc, that's how hazard ratios for experimental interventions work, they are just a multiplier for the number of events under SoC in order to give number of events under Exp care, so more events under SoC means more events under the experimental strategy when it's hazard ratio is applied to the number of events under SoC. In the model, the function is called in the sensitivity analysis and applies a hazard ratio  
  
# I also describe applying the Exp HR to the post changes SoC in my function where this is done.  
  
# I can't use Free to Progressed probabilities, because now those probabilities are time-sensitive, so if I try to change these here and include them in the TWSA I'll be including too many things, just like in the tornado diagram, as I explain in my description on hazard ratios.  
  
# Minimum\_p\_FP\_SoC <- p\_FP\_SoC - 0.20\*p\_FP\_SoC  
  
# Maximum\_p\_FP\_SoC <- p\_FP\_SoC + 0.20\*p\_FP\_SoC  
  
# Minimum\_p\_FP\_Exp <- p\_FP\_Exp - 0.20\*p\_FP\_Exp  
  
# Maximum\_p\_FP\_Exp <- p\_FP\_Exp + 0.20\*p\_FP\_Exp  
  
# To address this, I can apply this mean, max and min to the hazard ratios instead, knowing that when run\_owsa\_det is run in the sensitivity analysis it calls the oncology\_semi\_markov\_function to run and in this function the hazard ratios generate the survivor function, and then these survivor functions are used to generate the probabilities (which will be cycle dependent), so I am varying the transition probabilities by 20% using a static value.  
  
  
# Alterantive modelling approach I could try here if so inclined:  
  
  
# "When the base-case result of an analysis strongly favors one alternative, a threshold analysis may be presented as a worst-case or ''even if'' analysis (e.g., ''Even if the risk reduction is as low as X, the ICER remains below Y,'' or ''Even if the relative risk reduction with alternative A is as low as X and the cost of treatment is as high as Y, alternative A dominates B''). Threshold values can easily be combined with the tornado presentation by marking them on the horizontal bars."  
  
# <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>  
  
  
# If for some reason I wanted a three-way sensitivity analysis per Andrew Brigg's discussion on page 8 above, there may be some code to describe doing this here: https://rdrr.io/github/syzoekao/CEAutil/src/inst/rmd/Rcode.R (github here: https://github.com/syzoekao/CEAutil/) and here: https://syzoekao.github.io/CEAutil/#44\_two-way\_sensitivity\_analysis also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\introduction to R for cost-effectiveness analysis.pdf  
  
# Again, I think this is best suited to a NMB, where you will have 2 NMB values, one for EXP and one for SOC, for ICERAs, because it's a ratio between SoC and EXP you will only have one ICER value, and in the code, both SOC and EXP are assigned it, so in the diagram both colours will sit on top of eachother for SOC and EXP and also, even if changing one thing changes the ICER value, it is changing it for both SOC and EXP, because they both have the same ICER, so there won't be sections of one colour and not another.

# 09 Probabilistic Sensitivity Analysis (PSA)

If you wanted to calculate the proportion of samples for which the incremental net benefit is positive, or try Gianlucas way of plotting results as a cost-effectiveness plane, you could try BCEA here: hazaC:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Health Economic Modeling in R A Hands-on Introduction\Health-Eco\Markov models\markov\_smoking\_probabilistic (there’s also definitely good tutorials on using BCEA online) and he has slides on this here: <https://gianluca.statistica.it/slides/ispor-2022/bcea/#27>

Here’s how to generate the standard error (SE) for a log-normal distribution, and then to make random draws for a log-normal distribution: <https://hesim-dev.github.io/hesim/articles/markov-cohort.html> also here: <https://devinincerti.com/2018/02/10/psa.html#beta-distribution>

The best way to check if the way you are calculating distributions is correct is by making comparisons to publications which include distributions, ses, etc., here are some publications that include that information:

This study has an appendix where they report distribution, SE, base case, range and alpha and beta for the beta distribution and normal distribution: Spackman, D. E., & Veenstra, D. L. (2008). A cost-effectiveness analysis of currently approved treatments for HBeAg-positive chronic hepatitis B. Pharmacoeconomics, 26(11), 937-949. <https://sci-hub.ru/10.2165/00019053-200826110-00006> also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\spackman2008.pdf

Appendix here: <http://download.lww.com/wolterskluwer_vitalstream_com/permalink/pcz/a/00019053-920082611-00002.pdf> and also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\00019053-920082611-00002.pdf

[It also may be worth following some of their writing style.]

This also has mean, higher and lower CI and SE and distribution: <https://www.ncbi.nlm.nih.gov/books/NBK97497/pdf/Bookshelf_NBK97497.pdf> also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\Bookshelf\_NBK97497.pdf

PDF Page 212 of: Picot, J., et al. “The clinical effectiveness and cost-effectiveness of bortezomib and thalidomide in combination regimens with an alkylating agent and a corticosteroid for the first-line treatment of multiple myeloma: a systematic review and economic evaluation.” Health technology assessment (Winchester, England) 15.41 (2011): 1. <https://www.ncbi.nlm.nih.gov/books/NBK97497/pdf/Bookshelf_NBK97497.pdf>

This study has mean and SE so you may be able to find the range around the mean in the studies they cite: Is it worth offering a routine laparoscopic cholecystectomy in developing countries? A Thailand case study <https://sci-hub.ru/10.1186/1478-7547-3-10>

Sharp, Linda, et al. “Cost-effectiveness of population-based screening for colorectal cancer: a comparison of guaiac-based faecal occult blood testing, faecal immunochemical testing and flexible sigmoidoscopy.” British journal of cancer 106.5 (2012): 805-816. <https://www.nature.com/articles/bjc2011580.pdf> also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\bjc2011580.pdf

Koen has an excel and paper here, but it’s probably more useful for point estimates: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Matching the model with the evidence Koen both are saved here, but the filepath may be too long to access them without moving them to downloads or desktop.

This has mean and alpha beta but nothing else: Kristin, Erna, et al. “Economic evaluation of adding bevacizumab to chemotherapy for metastatic colorectal cancer (mCRC) patients in Indonesia.” Asian Pacific Journal of Cancer Prevention: APJCP 22.6 (2021): 1921. <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Evidence%20Synthesis/Economic%20Models/Kristin%20et%20al_2021_Economic%20Evaluation%20of%20Adding%20Bevacizumab%20to%20Chemotherapy%20for%20Metastatic.pdf>

The information provided here is less infromative: Jenks, Michelle, et al. “Tegaderm CHG IV securement dressing for central venous and arterial catheter insertion sites: a NICE medical technology guidance.” Applied Health Economics and Health Policy 14.2 (2016): 135-149. <https://link.springer.com/content/pdf/10.1007/s40258-015-0202-5.pdf>

This may also have informations on SE’s given point estimates and 95 CI’s Baio, G. (2014). Bayesian models for cost‐effectiveness analysis in the presence of structural zero costs. Statistics in medicine, 33(11), 1900-1913. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4285321/pdf/sim0033-1900.pdf>

Obviously the Goldstein study, but this is more making use of point estiamtes: <file:///C:/Users/Jonathan/Dropbox/PC/Downloads/goldstein2014.pdf>

This may have some limited info on mean, range and SE in the Tables at the end: “How sensitive are cost-effectiveness analyses to choice of parametric distributions?” <https://www.mrc-bsu.cam.ac.uk/wp-content/uploads/CE_para_york.pdf>

There’s information on mean alpha beta and se here: Cost-effectiveness of insulin pumps compared with multiple daily injections both provided with structured education for adults with type 1 diabetes: a health economic analysis of the Relative Effectiveness of Pumps over Structured Education (REPOSE) randomised controlled trial <https://bmjopen.bmj.com/content/bmjopen/8/4/e016766.full.pdf>

This may also have some information: Cost-effectiveness and programmatic benefits of maternal vaccination against pertussis in England <https://sci-hub.ru/10.1016/j.jinf.2016.04.012>

This has a point estimate, range, distribution and alpha beta: Campbell, J. R., Johnston, J. C., Cook, V. J., Sadatsafavi, M., Elwood, R. K., & Marra, F. (2019). Cost-effectiveness of latent tuberculosis infection screening before immigration to low-incidence countries. Emerging infectious diseases, 25(4), 661. <https://sci-hub.ru/10.3201/eid2504.171630>

This has a point estimate, range, distribution, alpha and beta: Gao, L., Moodie, M., Mitchell, P. J., Churilov, L., Kleinig, T. J., Yassi, N., … & EXTEND-IA TNK Investigators. (2020). Cost-effectiveness of tenecteplase before thrombectomy for ischemic stroke. Stroke, 51(12), 3681-3689. <https://www.ahajournals.org/doi/epdf/10.1161/STROKEAHA.120.029666>

There is information on point estimates, se, alpha and beta on PDF Page 71 of :A cluster randomised trial, cost-effectiveness analysis and psychosocial evaluation of insulin pump therapy compared with multiple injections during flexible intensive insulin therapy for type 1 diabetes: the REPOSE Trial <https://www.journalslibrary.nihr.ac.uk/hta/hta21200#/abstract>

Slide 39 of this has a mean, se, distribution, low and high 95% interval: <https://clas.ucdenver.edu/marcelo-perraillon/sites/default/files/attached-files/lecture_13_uncertainty.pdf> also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\lecture\_13\_uncertainty.pdf

This has mean, range, shape, scale and distribution for Beta and Log-normal: Appendix F: Health economics report (2015 work undertaken by NICE Internal Clinical Guidelines) <https://www.nice.org.uk/guidance/ng33/evidence/appendix-f-full-he-report-pdf-80851860763>

It’s possible that this will have some information for normal distributions: Bayesian sample size determination for costeffectiveness studies with censored data <https://sci-hub.ru/10.1371/journal.pone.0190422#savedHeader>

Beta and Gamma with point estimate and shape and scale: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\paper with shapes and scales\12874\_2017\_437\_MOESM1\_ESM.pdf

This book explains some of the concepts you need to describe, like how to read a two-way sensitivity analysis graph: Muennig, P., & Bounthavong, M. (2016). Cost-effectiveness analysis in health: a practical approach. John Wiley & Sons. - saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\cost effectiveness a.pdf

There’s some code for using the method of moments to generate parameters for drawing random samples here: <https://devinincerti.com/2018/02/10/psa.html> also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\Probabilistic Sensitivity Analysis with R.pdf

formulas to calculate alpha and beta for gamma and beta distriobutions appear in the literature here: Briggs, Andrew H., et al. “Probabilistic analysis of cost-effectiveness models: choosing between treatment strategies for gastroesophageal reflux disease.” Medical decision making 22.4 (2002): 290-308. www.med.mcgill.ca/epidemiology/courses/EPIB654/Summer2010/EF/example%20PPI.pdf

This may provide some useful guides on the PSA <https://rdrr.io/github/DARTH-git/dampack/api/>

Go back to the resources and papers you reviewed for the deterministic sensitivity analysis.

In probabilistic sensitivity analysis, parameters were varied simultaneously according to their sampling distributions. We used g distribution for cost parameters, b distribution for parameters bounded on the 0 to 1 interval (eg, AE incidence estimates and utilities), and the uniform distribution for the discount factor. The range for the discount factor was 0.00 to 0.05. In multivariate analysis, we ran 10,000 replications in the probabilistic sensitivity analysis. The baseline values, ranges, and distributions for model parameters are presented in Table 4. Per: Cost Effectiveness Analysis of Pharmacokinetically-Guided 5-Fluorouracil in FOLFOX Chemotherapy for Metastatic Colo . <file:///C:/Users/jonathanbriody/Downloads/goldstein2014.pdf>

This has some information on distributions:

<https://cran.r-project.org/web/packages/dampack/vignettes/psa_generation.html>

Also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\sensitivity analysis help files\Probabilistic Sensitivity Analysis\_ Generation.pdf

If we want to repeat all our one-way sensivity analysis probabilistically, rather than deterministically, you can see how to do this here, it might make for a good appendix because the description is technically complex, so would read well in the paper, but the application in R is actually very straightforward: <https://cran.r-project.org/web/packages/dampack/vignettes/psa_analysis.html>

**My writing on what a PSA is (then I’ll describe each results section):**

To evaluate the robustness of the model, we conducted a probabilistic sensitivity analysis (PSA) to determine the consequences of parameter uncertainty *in our model* for cost-effectiveness outcomes. Simultaneous samples were drawn from model parameters in Monte Carlo simulations according to their sampling distributions, reflecting any uncertainty in parameter estimates in the evidence. Thus, joint uncertainty was propagated through the model over 10,000 replications, providing a comprehensive assessment of model parameter uncertainty. Confidence intervals or standard errors extracted from published sources were used for the probabilities, adverse events, costs and utilities. Where this information was not available, model inputs were varied 20% above and below base case values per Flowers I think?. The baseline values, ranges, and distributions for model parameters are reported in Table X.

For the hazard ratio we generate random draws from a log-normal distribution. The value of ratio intervention effects have a lowest possible value of 0, a value of 1 corresponding to no intervention effect, and a highest value possible value of infinity. Logically, a ratio that halves effects and one that doubles effects should average to no effect. However, the average of 0.5 and 2 is 1.25, rather than 1. Thus, a log transformation is applied to make the hazard ratio number scale symmetric: the log of 0 is minus infinity, the log of 1 is zero, and the log of infinity is infinity. The log of 0.5 is –0.69 and the log of the OR of 2 is 0.69, such that when the halving and doubling ratios are averaged, a value of 0 is obtained, i.e., the log transformed value of 1, or no effect.

### PROBABILISTIC ANALYSIS ----  
  
# We conduct a probabilistic analysis (PSA) of the model to estimate the uncertainty in the model outcomes.  
  
# To do this, we generate samples for each model parameter from parametric distributions and evaluate the model for each set of parameter samples.  
  
  
## Sampling the parameter values ----  
  
# In order for the results reported in the PSA to be reproducible we need to set the random number seed. We also defining the number of runs, i.e., how many times we will re-sample for the parameter distributions, typically the number chosen is 10,000 in published studies, so we do that too:  
n\_runs <- 10000  
set.seed(123)  
  
# In the PSA, first we sample the parameter values from their distributions and store them in a data.frame.  
# If a model parameter has no uncertainty and, hence, is fixed, we can set it equal to itself.  
  
  
# To take random draws for the Weibull parametric survival distribution that we fitted at the start of the document using the 'flexsurv' package, we use the following piece of code.  
  
# This applies mvrnorm (i.e., takes samples from the specified multivariate normal distribution) to the mean (mu) of   
  
# l\_TTP\_SoC\_weibull$coefficients  
# shape scale   
# 0.7700246 1.7425343   
  
# i.e., to the mean of the shape and the scale variables from the Weibull parametric survival distribution that we fitted.  
  
# And gives a sigma (a positive-definite symmetric matrix specifying the covariance matrix of the variables) that is equal to the covariance between the shape and the scale variables:  
  
# > l\_TTP\_SoC\_weibull$cov  
# shape scale  
# shape 0.001983267 0.000291494  
# scale 0.000291494 0.000782051  
  
# You'll see that it's covariance between the two variables because you can match shape with scale twice, and each time it's the same value, so the same covariance between the two variables.  
  
# n is just the number of samples.  
  
# Described here:  
# rdocumentation.org/packages/MASS/versions/7.3-58.1/topics/mvrnorm  
  
# So, correlated sets (a "set" here is like one random draw from the distribution, because the draw is for shape and scale at the same time) of coefficients for the survival distribution (i.e., correlated sets of shape and scale values for the survival distribution, which are the coefficients as above) are generated using the variance-covariance matrix that was obtained from fitting the weibull distribution (the cov in l\_TTP\_SoC\_weibull is equal to the covariance between the shape and the scale variables and is printed as a 2 by 2 matrix above) and a multivariate normal distribution (this is just saying we are applying mvrnorm to take samples from the multivariate normal distribution for the mean (mu) values of shape and scale).  
  
# Multivariate normal distribution described here too: https://devinincerti.com/2018/02/10/psa.html'  
  
  
m\_coef\_weibull\_SoC <- mvrnorm(  
 n = n\_runs,   
 mu = l\_TTP\_SoC\_weibull$coefficients,   
 Sigma = l\_TTP\_SoC\_weibull$cov  
)  
  
head(m\_coef\_weibull\_SoC)

## shape scale  
## [1,] 0.4295 5.606  
## [2,] 0.3354 5.674  
## [3,] 0.2912 5.562  
## [4,] 0.3096 5.674  
## [5,] 0.3319 5.646  
## [6,] 0.2907 5.549

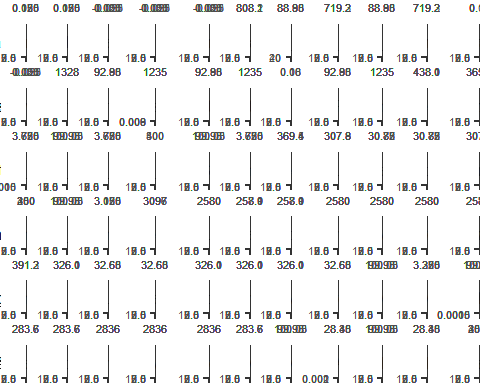
m\_coef\_weibull\_OS\_SoC <- mvrnorm(  
 n = n\_runs,   
 mu = l\_TTD\_SoC\_weibull$coefficients,   
 Sigma = l\_TTD\_SoC\_weibull$cov  
)  
  
head(m\_coef\_weibull\_OS\_SoC)

## shape scale  
## [1,] 0.3453 6.633  
## [2,] 0.3215 6.592  
## [3,] 0.4090 6.688  
## [4,] 0.3678 6.705  
## [5,] 0.3881 6.589  
## [6,] 0.3973 6.650

# Now that we have applied mvrnorm to get a random shape and a random scale we don't need to include HR\_FP\_SoC in our data.frame to create random probability draws for SoC, because we can just select the shape and scale from m\_coef\_weibull\_SoC so that in our function the updated coef\_weibull\_shape\_SoC and coef\_weibull\_scale\_SoC are used to generate S\_FP\_SoC with a random value, which will in turn generate transition probabilities under standard of care with a random value.  
  
  
  
  
  
  
  
  
  
  
 # According to:  
#  
# C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Health Economic Modeling in R A Hands-on Introduction\Health-Eco\Markov models\markov\_smoking\_probabilistic.R  
#   
#   
# It is OK to do the following:  
#   
# # QALY associated with 1-year in the smoking state is Normal(mean = 0\_95, SD = 0\_01)  
# # Divide by 2 as cycle length is 6 months  
# state\_qalys[, "Smoking"] <- rnorm(n\_samples, mean = 0.95, sd = 0.01) / 2  
#   
# So, when doing probabilistic sensitivity analysis and I need my mean and sd for the method of moments this may be useful information when drawing these from population norms.  
# This will also be useful for utility and PSA in the adverse event setting.  
  
  
# I create the SD to include in the sensitivity analysis applied to the hazard ratio below, as I need to create it outside the data.frame for the PSA:  
  
UpperCI <- 0.87  
LowerCI <- 0.53  
SD <- (log(UpperCI) - log(LowerCI)) / 3.92  
  
MEAN\_HR\_FP\_Exp <- HR\_FP\_Exp  
# I copy the hazard ratio above in case the below would start building HR\_FP\_Exp from the updated HR\_FP\_Exp made from random draws, i.e., in case it started including the HR\_FP\_Exp updated from the random draws from HR\_FP\_Exp in meanlog = (HR\_FP\_Exp), when we really need to be drawing the variability from the hazard ratio we started out with, and not a hazard ratio that we started out with, but that has since already been varied and will be varied from again if we take a random draw from this pre-varied hazard ratio.  
  
# HR\_FP\_Exp = rlnorm(n\_runs, meanlog = log(HR\_FP\_Exp), sdlog = SD)  
  
# !!! I MAY NEED TO REPEAT THIS FOR: HR\_PD\_Exp  
  
# I create the SD to include in the sensitivity analysis applied to the hazard ratio below, as I need to create it outside the data.frame for the PSA:  
  
OS\_UpperCI <- 0.86  
OS\_LowerCI <- 0.49  
OS\_SD <- (log(OS\_UpperCI) - log(OS\_LowerCI)) / 3.92  
  
MEAN\_HR\_PD\_Exp <- HR\_PD\_Exp  
# I copy the hazard ratio above in case the below would start building HR\_FP\_Exp from the updated HR\_FP\_Exp made from random draws, i.e., in case it started including the HR\_FP\_Exp updated from the random draws from HR\_FP\_Exp in meanlog = (HR\_FP\_Exp), when we really need to be drawing the variability from the hazard ratio we started out with, and not a hazard ratio that we started out with, but that has since already been varied and will be varied from again if we take a random draw from this pre-varied hazard ratio.  
  
  
  
  
# Below I create the data.frame that I will use in the PSA.  
# So, this is the PSA input dataset.  
  
df\_PA\_input <- data.frame(  
 coef\_weibull\_shape\_SoC = m\_coef\_weibull\_SoC[ , "shape"],  
 coef\_weibull\_scale\_SoC = m\_coef\_weibull\_SoC[ , "scale"],  
 coef\_TTD\_weibull\_shape\_SoC = m\_coef\_weibull\_OS\_SoC[ , "shape"],  
 coef\_TTD\_weibull\_scale\_SoC = m\_coef\_weibull\_OS\_SoC[ , "scale"],  
   
  
   
### Hazard Ratios   
   
# They use exp for rnorm (i.e., to make random draws from a normal distribution) I to use want rlnorm.  
  
# HR\_FP\_Exp = exp(rnorm(n\_runs, log(0.6), 0.08))  
  
# The rlnorm() function in R is used for generating random draws from a log-normal distribution.  
   
# To make draws from the log-normal distribution you need to enter a hazard ratio and standard deviation.  
  
# If you find a hazard ratio and confidence interval in the literature, rather than a hazard ratio and a standard deviation, you can make conversions to a standard deviation.  
  
# To generate random draws for the hazard ratio, I need a mean for the hazard ratio(just the hazard ratio value itself), and a standard deviation built from the 95% confidence interval of the hazard ratio (the SD is built from log(Upper CI) - log(Lower CI)/2\*SE, so sd is already built from logs for inclusion in rlnorm and doesnt need to be set as sdlog = log() unlike meanlog = log() where you need to take the log of the hazard ratio value).   
  
  
# SD: (natural log(Upper confidence interval) - natural log(lower confidence interval) / 2\*Standard error (i.e. 1.95\*2 = 3.92 for a 95% confidence interval)  
  
  
# So, it looks like I take the natural log of the upper limit minus the natural log of the lower limit (in confidence intervals the lower limit is reported on the left and the upper limit is reported on the right, so it would be 95% CI (30.0 [LOWER LIMIT], 34.2[UPPER LIMIT]), and I would rearrange these to have [ln(UPPER LIMIT) - ln(LOWER LIMIT)], i.e., [ln(34.2)-ln(30.0)]) and divide by 2 times the standard error. Provided the sample size is large, then for the 95% confidence interval this would be 2 x 1.96 = 3.92 For 90% confidence intervals 3.92 should be replaced by 3.29, and for 99% confidence intervals it should be replaced by 5.15.   
  
# I often come across hazard ratios and their confidence intervals in the published literature on clinical trials, but rarely do I see standard deviations.   
   
# A typical example from the literature is the following: "HR, 0.69; 95% CI, 0.54 to 0.89 in mCRC for cetuximab plus FOLFOX-4 vs FOLFOX-4 alone" - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7044820/pdf/bmjopen-2019-030738.pdf:  
#   
# To compute the standard deviation for hazard ratios from the above typical reports in the literature, so that I can then generate random draws from a log-normal distribution for a hazard ratio I would do the following:  
#   
# (natural log(Upper confidence interval) - natural log(lower confidence interval) / 2\*Standard error (i.e. 1.95\*2 = 3.92 for a 95% confidence interval)  
#   
# or in R: (log(0.89) - log(0.54)) / 3.92 = 0.1274623  
#   
#   
# UpperCI <- 0.89  
# LowerCI <- 0.54  
# SD <- (log(UpperCI) - log(LowerCI)) / 3.92  
# SD = 0.1274623  
#   
  
# If I need to provide a citation for this formula I can use the below which, although this does not provide the formula I use, it provides enough information that is similar to the above, and which I could say ultimately informed the formula:  
# Higgins, J. P. T., & Deeks, J. J. (2011). Chapter 7.7. 3.3: Obtaining standard deviations from standard errors, confidence intervals, t values and P values for differences in means. Cochrane handbook for systematic reviews of interventions, Version, 5(0).  
#   
#   
# This approach is also supported by the following post, which provides good clarity on all of the above:  
#   
# https://stats.stackexchange.com/questions/546192/calculate-the-standard-deviation-from-a-hazard-ratios-confidence-interval  
  
  
# random draws from a log-normal distribution  
  
# hr\_draws <- rlnorm(nsims, meanlog = log(mean), sdlog = SD).   
  
#   
# We take random draws from the log-normal distribution for ratios, because if we were to take random draws from the ratio as it stands, because the hazard ratio can't go any lower than 0 but can go to plus infinity, our random draws would be skewed in the values we take from the distribution, on the other hand, if we were to put things on the natural log scale we would be taking our random draws from a more normalised distribution. So, you'll see that in:  
#   
# rlnorm(n\_sim, meanlog = log(mean), sdlog = SD)  
#   
# it's the log of the mean (the ratio value, so the hazard ratio) and the log of the standard deviation (which we put on the log scale when we calculate it so we don't need to put in as sdlog = log() here, just sdlog = ) in order to be taking random draws from a log-normal distribution (i.e. a log normally distributed HAZARD RATIO).  
  
HR\_FP\_Exp = rlnorm(n\_runs, meanlog = log(MEAN\_HR\_FP\_Exp), sdlog = SD),  
HR\_PD\_Exp = rlnorm(n\_runs, meanlog = log(MEAN\_HR\_PD\_Exp), sdlog = OS\_SD),  
  
  
# Per [SLIDE 16] onwards in C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\notes.txt I don't think we need to worry about the duration of time a hazard ratio refers to. In my model I definitely don't, as the hazard ratio and underlying rate survival curves that make up SoC transition probabilities come from the same paper, and thus the same time period.   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
#### To generate random draws for the probabilities:  
  
# IMPORTANT, THIS VALUE IS BOUNDED BY 1 OR 0:  
# When calculating se from point estimates, remember that utility and probability values cannot be less than 0 or greater than 1, so don't calculate a min or max that is less than 0 or greater than 1. If you do, then change it to be 0 or 1, as appropriate, i.e., rather than 1.02, make it 1.  
  
# When, as in slide 12 of C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\a212ProbabilisticDistributions-210604-154.pdf you can have many transitions from one state into others [i.e. in A you can stay in State A or you can go into State B, State C or State D rather than just going from state C into State D] it's not clear how you can do this in the beta code.  
  
# i.e., when alpha values are all the times things happened, while beta values are all the times they didnt, how do you apply   
  
# transistion\_probabiliy <- rbeta(n\_runs,alpha,beta)  
# when there are so many things that can happen?  
  
# FOR NON-MULTI STATE, I.E. FOR THE SICK TO DEAD STATE IN THE MARKOV MODEL YOU CAN APPLY THE BETA DISTRIBUTION, BECAUSE YOU CAN ONLY LEAVE (ALPHA) OR STAY (BETA).  
  
# Sometimes a Dirichlet is suggested, a Dirichlet distribution is the multidimensional generalization of the beta distribution, so it's the same as applying the beta distribution, - as per: https://www.rdocumentation.org/packages/rBeta2009/versions/1.0/topics/rdirichlet but I would need to have all the counts included I think, which works when you are drawing all your transition data from one larger dataset, like in a clinical trial, but my data necessarily comes from several sources, so I if I just build everything as conditional probabilities, per my notes in this Notepad [search 19/08/22: in C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\notes.txt] in combination with the slides [C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\a212ProbabilisticDistributions-210604-154.pdf] these are notes on, that is 1-Probabilities like I've done in R already, then I should be able to apply a Beta distribution to each probability and not have to worry about applying a Dirichlet.  
  
# Here's a quote explaining how best to do that from my notes above:  
  
# "In the above you would have a beta distribution with parameters 731, 527 [the 512+15] to look at whether you stayed in State B and then, conditional on leaving State B, we could assign another beta distribution with parameters 512 and 15 to whether you, conditional on having left State B then transited to State C. And, of course, death is then the third category. Death is the residual probability. [i.e., conditional on having left state B is 1-the probability of staying in State B, so you could apply a Beta distribution to the probability of going into State C multiplied by 1-the probability of staying in State B. Death would be 1-the probability you create of moving into State C, because if you're not going into State C then you're going into Death as there's nowhere else for you to go, I think if you wanted to apply a Beta distribution to Death you could apply a Beta distribution to the probability of going into Death and then multiply that by 1-the probability of staying in the B state]."  
  
# Conditional probabilities are built from the below p\_'s in the function, so you basically create the beta distribution probabilities below as p\_'s, and then in the function you multiply them by 1-whatever probability as necessary such that they become conditional probabilities.  
  
##1  
  
# Fitting Beta distributions to (constant) probability parameters [I guess the "constant" here is to clarify that this isnt how you do it for time varying transition probabilities, like the transition probabilities I create from Weibull].  
  
# There are two main ways to do this, per: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.3 Practical exercise R\A231\_Making\_Models\_Proba.pdf say you had something like the following printed in a study:  
  
# ‘The hospital records of a sample of 100 patients receiving a primary THR were examined retrospectively. Of these patients, two patients died either during or immediately following the procedure. The operative mortality for the procedure is therefore estimated to be 2%.’  
  
# You could use this information directly to specify the parameters (alpha and beta) of a beta distribution for the probability of operative mortality during primary surgery. These are noted by “a.” and “b.” respectively. Your alpha values are all the times things happened, while your beta values are all the times they didnt, so 2 and 98.  
  
# alpha <- 2 ## alpha value for operative mortality from primary surgery  
# beta <- 100- alpha ## beta value for operative mortality from primary surgery  
  
# tp.PTHR2dead <- rbeta(n\_runs,alpha,beta) ## Operative mortality rate (OMR) following primary THR  
  
# (as coded in C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.3 Practical exercise R\A233\_Making\_Models\_Probabilistic\_Solutions-210604 (1))  
  
# The question is, does this give the transition probability for the time period that the quote describes? Well in the same pdf and r code as above you have the following:  
  
# "The following information has been provided to you concerning revision procedures among patients already having had a revision operation (i.e. the re-revision risk). ‘The hospital records of a sample of 100 patients having experienced a revision procedure to replace a failed primary THR were reviewed at one year. During this time, four patients had undergone a further revision procedure.’iv) Use this information to fit a constant transition probability for the annual re-revision rate parameter (tp.rrr), using the same approach as for operative mortality above. Your beta value for re-revision risk should be equivalent to 96."  
  
  
# a.rrr <- 4 ## alpha value for re-revision risk  
# b.rrr <- 100-a.rrr ## beta value for re-revision risk  
# tp.rrr <-rbeta(1,a.rrr,b.rrr) ## Re-revision risk transition probability  
  
# tp.PTHR2dead  
# tp.RTHR2dead  
# tp.rrr  
  
  
# Because the cycle length in the model is one year according to page 3 of C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A1\_Advanced Markov Modelling\A1.3 Practical exercise R\A131\_Advanced\_Markov\_Modelling\_Instructions-210528-18 this indicates that it does.  
  
  
##2  
 # METHOD OF MOMENTS:  
  
  
# In cases where you don't have this information, but you do have a mean and a standard error, you can do the following (per: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\notes.txt):  
  
# My initial reading for probabilistic analysis with probabilities suggested that the method of moments can be applied to a mean and standard error for a transition probability sourced from the literature to create an appropriate alpha and beta to use in a probabilistic sensitivity analysis:  
  
# So, I built the following code to translate the formula for the methods of moments into R, to allow us to take our transition probabilities from the literature:  
#   
#   
# ## Methods of moments  
#   
#   
# #Assume the below mean and std error come from the reported literature  
#   
# mean <- 0.75 ## mean from the literature  
# std.error <- 0.04 ## standard error from the literature  
#   
# # The method of moments can be coded up as below to get the alpha and beta from the sample using the mean and standard error from the sample:  
#   
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
#   
# # Following this, you can randomly draw from the beta distribution using the below, I only have one random draw, but you could change that 1 to 10,000 (the standard number of draws) and save these as a matrix of draws to use in your probabilistic sensitivity analysis:  
#   
# probabilistic.beta.dist.draw <- rbeta(1,alpha,beta) ## drawing from the Beta distribution based on a and b  
  
# Per the above pdf:  
  
# "Note that the rbeta function takes 3 arguments, first the number of draws or samples, the alpha parameter (named shape1) and the beta parameter (named shape2)."   
# "rbeta(n = 5, shape1 = 50, shape2 = 100)"  
  
# while in their R code they use the below, as included above, so I don't think it matters if you include shape1= or just include the alpha and beta directly, but I'll keep it as below, as that's how DARTH do it.   
  
# tp.rrr <-rbeta(1,a.rrr,b.rrr) ## Re-revision risk transition probability  
  
# The below gives the same results either way:  
  
# set.seed(100)  
# n = 1  
# b4 <- rbeta(n, shape1 = 2, shape2 = 8)  
# b4  
# [1] 0.1329533  
#   
# set.seed(100)  
# n = 1  
# b4 <- rbeta(n, 2, 8)  
# b4  
# [1] 0.1329533  
  
## 3  
  
# RANGE:  
  
# I don't describe generating random draws when the probabilities don't have an SE above, however, I can do this in the same way that I figured out how to do this for the utility beta distributions below. If they are reported with a range, he section on utility below also advises how to handle this.  
  
  
  
  
# Brigs code to generate the SE for a parameter with a range perfectly centered around the mean (briggsse), and then repeated but for the situations where the range isnt PERFECTLY centered around the mean (altbriggsse), i.e., the min is further away from the mean than the max, or vice versa (even a little):  
  
max <- 0.22,  
min <- 0.12,   
mean <- P\_OSD\_SoC,  
  
briggsse <- ((max)-(mean))/1.96,  
# altbriggsse <- (max-min)/(2\*1.96),  
  
## Generating the alpha and beta:  
  
std.error <- briggsse, ## briggsse OR altbriggsse - as appropriate  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha.plus.beta,  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
   
P\_OSD\_SoC = rbeta(n\_runs, shape1 = alpha, shape2 = beta),  
mean(P\_OSD\_SoC),  
P\_OSD\_SoC\_alpha <- alpha,  
P\_OSD\_SoC\_beta <- beta,  
  
  
  
max <- 0.22,  
min <- 0.12,   
mean <- P\_OSD\_Exp,  
  
briggsse <- ((max)-(mean))/1.96,  
# altbriggsse <- (max-min)/(2\*1.96),  
  
## Generating the alpha and beta:  
  
std.error <- briggsse, ## briggsse OR altbriggsse - as appropriate  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha.plus.beta,  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
P\_OSD\_Exp = rbeta(n\_runs, shape1 = alpha, shape2 = beta),  
mean(P\_OSD\_Exp),  
P\_OSD\_Exp\_alpha <- alpha,  
P\_OSD\_Exp\_beta <- beta,  
  
  
  
  
mean<- p\_FA1\_STD,  
Maximum <- Maximum\_p\_FA1\_STD,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
p\_FA1\_STD = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_p\_FA1\_STD <- alpha,  
beta\_p\_FA1\_STD <- beta,  
  
  
  
mean<- p\_FA2\_STD,  
Maximum <- Maximum\_p\_FA2\_STD,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
p\_FA2\_STD = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_p\_FA2\_STD <- alpha,  
beta\_p\_FA2\_STD <- beta,  
  
  
  
mean<- p\_FA3\_STD,  
Maximum <- Maximum\_p\_FA3\_STD,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
p\_FA3\_STD = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_p\_FA3\_STD <- alpha,  
beta\_p\_FA3\_STD <- beta,  
  
  
  
mean<- p\_FA1\_EXPR,  
Maximum <- Maximum\_p\_FA1\_EXPR,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
p\_FA1\_EXPR = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_p\_FA1\_EXPR <- alpha,  
beta\_p\_FA1\_EXPR <- beta,  
  
  
  
mean<- p\_FA2\_EXPR,  
Maximum <- Maximum\_p\_FA2\_EXPR,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
p\_FA2\_EXPR = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_p\_FA2\_EXPR <- alpha,  
beta\_p\_FA2\_EXPR <- beta,  
  
  
  
mean<- p\_FA3\_EXPR,  
Maximum <- Maximum\_p\_FA3\_EXPR,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
p\_FA3\_EXPR = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_p\_FA3\_EXPR <- alpha,  
beta\_p\_FA3\_EXPR <- beta,  
  
  
  
  
  
  
   
  
# If I was going with my initial plan of applying the PSA to the original COLOSSUS limited sample size, I could say, the uncertainty in this limited sample size is propogated through the model by applying a dirichlet and beta distribution to a fully probabilistic analysis of the data.  
  
  
# Calculate the mean, maximum and variance of the Beta and Gamma here: https://www.pluralsight.com/guides/beta-and-gamma-function-implementation-in-r also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\R Guide\_ Beta and Gamma Function Implementation \_ Pluralsight.pdf  
  
  
  
  
 # Cost vectors with length n\_runs  
  
  
## Costs  
  
  
# 1) COSTS WITH A RANGE:  
  
  
  
# Maximum <- SOMEVALUEHERE  
# Mean <- SOMEVALUEHERE  
  
# se <- ((Maximum) - (Mean)) / 2  
# se   
   
# mean <-Mean  
# mean  
   
# mn.cIntervention <- mean ## mean cost of intervention  
# se.cIntervention <- se ## standard error of cost of intervention  
   
# a.cIntervention <- (mn.cIntervention/se.cIntervention)^2 ## alpha value for cost of intervention (shape)  
# b.cIntervention <- (se.cIntervention^2)/mn.cIntervention ## beta value for cost of intervention (scale)  
   
# a.cIntervention  
# b.cIntervention  
   
   
# c\_H = rgamma(n\_sim, shape = a.cIntervention, scale = b.cIntervention), # cost of intervention  
   
# If costs are reported with a 95% confidence interval the maximum is the upper confidence interval. If the interval is not centered on the mean, use the maximum or minimum in place of the upper confidence interval, depending on which one is further away from the mean SE = (m-CIlower)/t, here the bit in the brackets is rearranged when dealing with lower limits to have the MEAN minus the lower confidence interval so that things don't become negative (because the lower value will of course always be lower than the mean, as it's supposed to be below the mean, so a smaller number minus a larger number will be negative), but when the interval is centered on the mean the formula of SE = (CIupper-m)/t and SE = (m-CIlower)/t are inherently doing the same thing, that is getting the difference between the mean and the confidence interval and dividing this by t. So both formulas are options to use in getting the SE, and an author may choose to use the one which incorporates the interval that is farthest from the mean in order to incorporate as much variability in doing this SE calculation as possible, i.e. to reflect the variability in their SE calculation.  
  
#   
# As per: https://stats.stackexchange.com/questions/550293/how-to-calculate-standard-error-given-mean-and-confidence-interval-for-a-gamma-d/550892#550892  
   
   
# "I would like to note that, while those values in the table happen to correspond with ±2σ, the minimum and maximum values do not generally follow such simple formula with mean plus-minus some standard deviation.  
   
# In this case, the minimum and maximum values only correspond to the interval μ±2σ because the distribution seems to have been truncated at those values."  
   
   
# i.e., my formula will only work if the minimum and maximum values reported in a study are ±2 SE from the mean, i.e. + 2 SE from the mean for the maximum value, and -2 SE from the mean for the minimum value.  
   
# And that will be the case any time there is a 95% Confidence Interval:  
   
  
# [ "Since 95% of values fall within two standard deviations of the mean according to the 68-95-99.7 Rule, simply add and subtract two standard deviations from the mean in order to obtain the 95% confidence interval. Notice that with higher confidence levels the confidence interval gets large so there is less precision." https://www.westga.edu/academics/research/vrc/assets/docs/confidence\_intervals\_notes.pdf also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\gammadist\confidence\_intervals\_notes.pdf  
  
# A good image can be found as per: https://www.quora.com/What-is-meant-by-one-standard-deviation-away-from-the-mean also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\gammadist\www-quora-com-What-is-meant-by-.pdf]  
  
# So, if I'm taking my costs and a range about these costs from the literature, I just need to make sure the range is a 95% Confidence Interval, like that reported here: Tilson, L., Sharp, L., Usher, C., Walsh, C., Whyte, S., O’Ceilleachair, A., ... & Barry, M. (2012). Cost of care for colorectal cancer in Ireland: a health care payer perspective. The European journal of health economics, 13(4), 511-524. (https://www.researchgate.net/publication/51188456\_Cost\_of\_care\_for\_colorectal\_cancer\_in\_Ireland\_A\_health\_care\_payer\_perspective) reports a cost in Table 3 for Colorectal cancer with a 95% CI: € 48,835 (€40,548–€62,582)..  
  
# For "PK test cost MEAN: 400.00 MIN: 300.00 MAX: 500.00" from Table 4, row 3 page 222 in Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., ... & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. Clinical colorectal cancer, 13(4), 219-225 plugged into the formula above to find the SE, I get an SE of 50, so plugging the 50 and mean into the CI = m ± t\*SE from Jochen above I get:  
   
# 400 + 2\*50 = 500  
   
# 400 - 2\*50 = 300  
   
# Which is the same confidence interval as I got in the Table 4 row 3, i.e. 300 and 500.  
   
# So, this might be a good way to check if my standard error is correct. Although this will only work if my confidence interval is centred on the mean, i.e. if I am the same distance from the mean to the max and to the min. If my confidence interval is NOT centred on the mean, I can calculate 2 standard errors, one between the lower interval and the mean, and one between the upper interval and the mean, and then apply these in CI = m ± t\*SE, knowing that:  
  
# CIupper = m + t\*SE  
  
# CIlower = m - t\*SE  
  
# I've figured out a simple test for whether the max/min reported is ±2σ:  
# se <- ((Max) - (Mean)) / 2  
# MaxMatch <- Mean + 2\*se   
# MinMatch <- Mean - 2\*se  
   
# I can then check if the max/min reported in a Table match the above, and if so I know that the max/min reported is ±2σ.  
   
# If not, it may be that they used the se off the minimum as it's further away, so try the below:  
   
# se <- ((Mean) - (Min)) / 2  
# MaxMatch <- Mean + 2\*se   
# MinMatch <- Mean - 2\*se  
   
   
# I can then check if the max/min reported in a Table match the above, and if so I know that the max/min reported is ±2σ.  
   
   
   
   
# Even if it's not symmetric, this may still be a 95% confidence interval if the mean is at 2 of one of the standard errors from the min or the max, we can see this in action below:  
   
   
# > Maximum <- 27655.84  
# > Mean <- 11565.60  
# >   
# > se <- ((Maximum) - (Mean)) / 2  
# > se   
# [1] 8045.12  
# >   
# > MaxMatch <- Mean + 2\*se   
# > MinMatch <- Mean - 2\*se  
# > MaxMatch  
# [1] 27655.84  
# > MinMatch  
# [1] -4524.64  
   
# But still, for these values  
   
# FN: Value: 11,565.60 Minimum: 7092.16 Maximum: 27,655.84 Distribution: γ: (2.067, 5596.247)  
   
# that are not centered on the mean, this still came out the same as the Table using this se <- ((Maximum) - (Mean)) / 2 method. Because looking at the below, the upper interval (the maximum) is further from the mean at a difference of 16090.24, than the lower interval (the minimum) at a difference between it and the mean of 4473.44 (i.e. the maximum is nearly four times further away from the mean that the minimum).  
   
   
# > Maximum <- 27655.84  
# > Minimum <- 7092.16  
# > Mean <- 11565.60  
   
# > Maximum - Mean  
# [1] 16090.24  
   
# > Mean - Minimum  
# [1] 4473.44  
   
   
# And when we plug the se in for the further away max we get:  
   
   
# > a.cIntervention  
# [1] 2.066671  
# > b.cIntervention  
# [1] 5596.247  
   
# Just the same as the values for FN above.  
#   
  
# So, this kind of applies to situations where an interval is reported, when you might want to check if it's a 95% confidence interval, and if not, which interval is furthest from the mean, in cases of a point estimate only, you do the following:  
  
  
  
  
# 2) POINT ESTIMATE ONLY:  
  
# If they arent reported with a 95% confidence interval, the Goldstein paper says: "Drug costs were varied within +/-20% of their baseline values as previously done by Goulart and Ramsey".  
   
# I checked this for FOLFOX drug cost Min: 443.91 Value: 355.13 Max: 532.69 gamma(100, 4.439)  
   
# Mean<- 443.91  
# Maximum <- Mean + 0.20\*Mean  
# Minimum <- Mean - 0.20\*Mean  
   
# Taking 20% plus the mean and 20% minus the mean, I get the same maximum and minimum as above, exactly:  
   
# > Maximum  
# [1] 532.692  
# > Minimum  
# [1] 355.128  
   
# They still applied the SE = (CIupper-m)/t or SE = (m-CIlower)/t to this interval which they generated from 20% plus the mean and 20% minus the mean.  
  
# In the Goldstein 2014 paper, where the ranges are more than 20% away from the mean, i.e., 25%, etc., they use the same formula for ranges that are symmetrically far from the mean, i.e., 25% bigger or smaller than the mean, and they use the same formula but incoporating the range that's furthest from the mean when generating the se when the ranges arent symmetric about the mean.  
  
  
  
  
  
# OK, so the question is, can we apply any percentage we like to the point estimate we find for costs to generate the max or min and then just apply the formula above to generate the SE? I reviewed Koen's study below:  
  
# Degeling, K., Franken, M. D., May, A. M., van Oijen, M. G., Koopman, M., Punt, C. J., ... & Koffijberg, H. (2018). Matching the model with the evidence: comparing discrete event simulation and state-transition modeling for time-to-event predictions in a cost-effectiveness analysis of treatment in metastatic colorectal cancer patients. Cancer epidemiology, 57, 60-67.  
  
# C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Matching the model with the evidence Koen  
  
# In the excel file, on the second sheet there is a row called parameter 1. For cost this parameter is the alpha value, and when you click on 16.000 it shows the formula that created this alpha value, which is the same as for my alpha value formula above. This formula includes the mean and the SE. When I take the mean (which is just the baseline value) and plug this into the R code to generate the SE:  
  
# I get the same se as in the excel file when making the max and min 50% larger or smaller than the mean in the code below.   
# Mean<- 2062.35  
# Maximum <- Mean + 0.50\*Mean  
# Minimum <- Mean - 0.50\*Mean  
# se <- ((Maximum) - (Mean)) / 2  
# se  
# 515.5875  
  
# Even more important, they apply the same code as I have for creating the alpha value above using this mean and SE generated just from taking 50% either side of the mean.   
  
# The implication of this is that you can generate the SE of a point estimate by taking any percentage either side of the mean and that you can also apply the code as above to generate the alpha and beta. Unfortunately in the excel file they don't create the beta in parameter 2, but the lambda, which is something else again.   
  
  
# This may solve the riddle of lambda in Koen's code, I noticed that here they use 1/Beta: https://www.pluralsight.com/guides/beta-and-gamma-function-implementation-in-r so I took that approach with the Lambda in Koen's code, plugging it in for beta, and got the same value of the mean as I started out with.  
  
# Mean<- 2062.35  
# Maximum <- Mean + 0.50\*Mean  
# Minimum <- Mean - 0.50\*Mean  
# se <- ((Maximum) - (Mean)) / 2  
  
# se  
# Maximum  
# Minimum  
# mean <-Mean  
  
# mn.cIntervention <- mean ## mean cost of intervention  
# se.cIntervention <- se ## standard error of cost of intervention  
# a.cIntervention <- (mn.cIntervention/se.cIntervention)^2 ## alpha value for cost of intervention (shape)  
# b.cIntervention <- (mn.cIntervention)/((se.cIntervention)^2) ## beta value for cost of intervention (scale) using Koen's backwards version of Andy's (se.cIntervention^2)/mn.cIntervention to generate the Lambda code.  
  
# a.cIntervention  
# b.cIntervention  
# se  
  
# c\_H = rgamma(10000, shape = a.cIntervention, scale = 1/b.cIntervention) # cost of intervention  
# mean(c\_H)  
   
  
# Likewise, if I do the following:  
  
# 1/ b.cIntervention  
# I get the exact same value for b.cIntervention as when I use the original code to generate b.cIntervention from Andy, as below.  
  
# b.cIntervention <- (se.cIntervention^2)/mn.cIntervention ## beta value for cost of intervention (scale)  
# b.cIntervention  
  
# So, it looks like the whole Lambda thing is just a different way of getting b.cIntervention, but Koen's way of doing it requires the addition of 1/ for the scale part of rgamma, while Andy's method doesnt.  
  
# So, in this it looks like Andy is generating the scale for the beta, while Koen is generating the rate:   
  
# This is supported by this:   
  
# "Density, distribution function, quantile function and random generation for the Gamma distribution with parameters alpha (or shape) and beta (or scale or 1/rate)." https://search.r-project.org/CRAN/refmans/Rlab/html/Gamma.html   
  
  
# "There is a R function for simulating this random variable. Here in addition to the number of values to simulate, we just need two parameters, one for the shape and one for either the rate or the scale. The rate is the inverse of the scale. The general formula is: rgamma(n, shape, rate = 1, scale = 1/rate)." https://pubs.wsb.wisc.edu/academics/analytics-using-r-2019/gamma-variables-optional.html also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\13.7 Gamma Variables (Optional) \_ Analytics Using R.pdf  
  
# I think the code above also implies that if you were to include it as a "rate" instead of calling it scale in the rcode for Koen's rate, then you wouldnt need to do the 1/ bit over scale. So, he may have taken this approach as well.This is supported by: "rate: a numeric vector in the range [0, Inf) that specifies the inverse of the scale parameter." https://docs.tibco.com/pub/enterprise-runtime-for-R/4.1.1/doc/html/Language\_Reference/stats/GammaDist.html  
  
# This is supported by the below:  
  
# > set.seed(1)  
# > x <- rgamma(n = 1000, shape = 5, scale = 3)  
# > mean(x)  
# [1] 14.61593  
# Knowing that the rate is the inverse of the scale, we can get the rate as 1/3 below:  
# > 1/3  
# [1] 0.3333333  
# > set.seed(1)  
# > x <- rgamma(n = 1000, shape = 5, rate = 0.3333333)  
# > mean(x)  
# [1] 14.61593  
# > set.seed(1)  
# > x <- rgamma(n = 1000, shape = 5, scale = 1/0.3333333)  
# > mean(x)  
# [1] 14.61593  
  
# Thus, Koen generated the rate and used that in rgamma, while Andy showed me how to use scale and use that in rgamma.   
  
  
# This 20% approach is also taken by Lesley tilson and cited in her study and other studies here: https://trello.com/c/m653JBWV/48-20-cost-sensitivity  
  
# administration\_cost  
Maximum <- Maximum\_administration\_cost,  
Mean <- administration\_cost,  
  
se <- ((Maximum) - (Mean)) / 2,  
se,   
  
mean <-Mean,  
mean,  
  
mn.cIntervention <- mean, ## mean cost of intervention  
se.cIntervention <- se, ## standard error of cost of intervention  
  
a.cIntervention <- (mn.cIntervention/se.cIntervention)^2, ## alpha value for cost of intervention (shape)  
b.cIntervention <- (se.cIntervention^2)/mn.cIntervention, ## beta value for cost of intervention (scale)  
  
a.cIntervention,  
b.cIntervention,  
  
administration\_cost = rgamma(n\_runs, shape = a.cIntervention, scale = b.cIntervention),   
  
a.cIntervention\_administration\_cost <-a.cIntervention,  
b.cIntervention\_administration\_cost <-b.cIntervention,   
  
  
# c\_PFS\_Folfox  
  
Maximum <- Maximum\_c\_PFS\_Folfox,  
Mean <- c\_PFS\_Folfox,  
  
se <- ((Maximum) - (Mean)) / 2,  
se,   
  
mean <-Mean,  
mean,  
  
mn.cIntervention <- mean, ## mean cost of intervention  
se.cIntervention <- se, ## standard error of cost of intervention  
  
a.cIntervention <- (mn.cIntervention/se.cIntervention)^2, ## alpha value for cost of intervention (shape)  
b.cIntervention <- (se.cIntervention^2)/mn.cIntervention, ## beta value for cost of intervention (scale)  
  
a.cIntervention,  
b.cIntervention,  
  
c\_PFS\_Folfox = rgamma(n\_runs, shape = a.cIntervention, scale = b.cIntervention),   
  
a.cIntervention\_c\_PFS\_Folfox <-a.cIntervention,  
b.cIntervention\_c\_PFS\_Folfox <-b.cIntervention,   
  
# c\_PFS\_Bevacizumab  
  
Maximum <- Maximum\_c\_PFS\_Bevacizumab,  
Mean <- c\_PFS\_Bevacizumab,  
  
se <- ((Maximum) - (Mean)) / 2,  
se,   
  
mean <-Mean,  
mean,  
  
mn.cIntervention <- mean, ## mean cost of intervention  
se.cIntervention <- se, ## standard error of cost of intervention  
  
a.cIntervention <- (mn.cIntervention/se.cIntervention)^2, ## alpha value for cost of intervention (shape)  
b.cIntervention <- (se.cIntervention^2)/mn.cIntervention, ## beta value for cost of intervention (scale)  
  
a.cIntervention,  
b.cIntervention,  
  
c\_PFS\_Bevacizumab = rgamma(n\_runs, shape = a.cIntervention, scale = b.cIntervention),   
  
a.cIntervention\_c\_PFS\_Bevacizumab <-a.cIntervention,  
b.cIntervention\_c\_PFS\_Bevacizumab <-b.cIntervention,   
  
  
# c\_OS\_Folfiri  
  
Maximum <- Maximum\_c\_OS\_Folfiri,  
Mean <- c\_OS\_Folfiri,  
  
se <- ((Maximum) - (Mean)) / 2,  
se,   
  
mean <-Mean,  
mean,  
  
mn.cIntervention <- mean, ## mean cost of intervention  
se.cIntervention <- se, ## standard error of cost of intervention  
  
a.cIntervention <- (mn.cIntervention/se.cIntervention)^2, ## alpha value for cost of intervention (shape)  
b.cIntervention <- (se.cIntervention^2)/mn.cIntervention, ## beta value for cost of intervention (scale)  
  
a.cIntervention,  
b.cIntervention,  
  
c\_OS\_Folfiri = rgamma(n\_runs, shape = a.cIntervention, scale = b.cIntervention),   
  
a.cIntervention\_c\_OS\_Folfiri <-a.cIntervention,  
b.cIntervention\_c\_OS\_Folfiri <-b.cIntervention,   
  
# The cost of being dead doesnt vary, it always costs the same to treat a dead person, which is 0 because we don't give any of our chemo, etc., to a dead body.  
 c\_D = 0,  
  
# c\_AE1  
  
Maximum <- Maximum\_c\_AE1,  
Mean <- c\_AE1,  
  
se <- ((Maximum) - (Mean)) / 2,  
se,   
  
mean <-Mean,  
mean,  
  
mn.cIntervention <- mean, ## mean cost of intervention  
se.cIntervention <- se, ## standard error of cost of intervention  
  
a.cIntervention <- (mn.cIntervention/se.cIntervention)^2, ## alpha value for cost of intervention (shape)  
b.cIntervention <- (se.cIntervention^2)/mn.cIntervention, ## beta value for cost of intervention (scale)  
  
a.cIntervention,  
b.cIntervention,  
  
c\_AE1 = rgamma(n\_runs, shape = a.cIntervention, scale = b.cIntervention),   
  
a.cIntervention\_c\_AE1 <-a.cIntervention,  
b.cIntervention\_c\_AE1 <-b.cIntervention,   
  
  
# c\_AE2  
  
Maximum <- Maximum\_c\_AE2,  
Mean <- c\_AE2,  
  
se <- ((Maximum) - (Mean)) / 2,  
se,   
  
mean <-Mean,  
mean,  
  
mn.cIntervention <- mean, ## mean cost of intervention  
se.cIntervention <- se, ## standard error of cost of intervention  
  
a.cIntervention <- (mn.cIntervention/se.cIntervention)^2, ## alpha value for cost of intervention (shape)  
b.cIntervention <- (se.cIntervention^2)/mn.cIntervention, ## beta value for cost of intervention (scale)  
  
a.cIntervention,  
b.cIntervention,  
  
c\_AE2 = rgamma(n\_runs, shape = a.cIntervention, scale = b.cIntervention),   
  
a.cIntervention\_c\_AE2 <-a.cIntervention,  
b.cIntervention\_c\_AE2 <-b.cIntervention,   
  
  
# c\_AE3  
  
Maximum <- Maximum\_c\_AE3,  
Mean <- c\_AE3,  
  
se <- ((Maximum) - (Mean)) / 2,  
se,   
  
mean <-Mean,  
mean,  
  
mn.cIntervention <- mean, ## mean cost of intervention  
se.cIntervention <- se, ## standard error of cost of intervention  
  
a.cIntervention <- (mn.cIntervention/se.cIntervention)^2, ## alpha value for cost of intervention (shape)  
b.cIntervention <- (se.cIntervention^2)/mn.cIntervention, ## beta value for cost of intervention (scale)  
  
a.cIntervention,  
b.cIntervention,  
  
c\_AE3 = rgamma(n\_runs, shape = a.cIntervention, scale = b.cIntervention),   
  
a.cIntervention\_c\_AE3 <-a.cIntervention,  
b.cIntervention\_c\_AE3 <-b.cIntervention,   
  
  
 # Utility vectors with length n\_runs   
  
# There may be helpful books here: C:\Users\Jonathan\Dropbox\PhD\HTA\Markov Modelling\books  
  
  
  
##### POINT ESTIMATES ONLY:  
  
  
  
# IMPORTANT, THIS VALUE IS BOUNDED BY 1 OR 0:  
# When calculating se from point estimates, remember that utility and probability values cannot be less than 0 or greater than 1, so don't calculate a min or max that is less than 0 or greater than 1. If you do, just round it to 0 or 1.  
  
  
# If you have a point estimate, then you can generate the standard error/standard deviation using the method of moments again.  
  
#Per Table 4 in the Goldstein paper:  
# > mean<- 0.850  
# > Maximum <- mean + 0.20\*mean  
# > Maximum  
# [1] 1.02  
# > Minimum <- mean - 0.20\*mean  
# > Minimum  
# [1] 0.68  
# > se <- ((Maximum) - (mean)) / 2  
# > se   
# [1] 0.085  
# > std.error <- se  
# > alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# > alpha <- mean\*alpha.plus.beta ## alpha (a)  
# > beta <- alpha\*(1-mean)/mean ## beta(b)  
# > alpha  
# [1] 14.15  
# > beta  
# [1] 2.497059  
# So, a perfect match to the Goldstein paper.  
# Because this approach to point estimates ends up making the same range around this point estimate as reported in Table 4, the implication is that when a point estimate is reported with a range, you can generate the SE the exact same way as above, that is, se <- ((Maximum) - (mean)) / 2, because we are applying the same manner of generating the SE to the range that was reported in the paper, as to the point estimate.  
  
  
# I double check the manner in which I calculate the SE with the excel file from Koen's paper: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Matching the model with the evidence Koen both are saved here, but the filepath may be too long to access them without moving them to downloads or desktop.  
  
# For his paper, I update the Maximum/Minimum part by taking the max as 50% bigger than the mean. When I do this, I get the exact same SE as he reports in the excel file, I know this because in his cells creating the alpha and beta values for utility or probability on sheet 2 of his excel file, he includes 2 numbers, one is the mean reported on sheet 1 for the parameter whose utility or probability he is taking random draws from, and then the other number is an EXACT match for the SE number I calculate, and must therefore be the SE number he calculates. Especially because he then applies a formula to this mean and SE to generate the alpha and the beta. You can see this in my model code below, where the mean and SE I report in my own analysis turn up in his alpha and beta formulas.  
  
# Although he uses a different method to calculate the alpha and the beta, these typically work out as very close to the values calculated for the alpha and beta using my method, which Briggs told me to use once I had the SE and mean, so I think Koen just has a deeper understanding of mathematics and is using this to apply a different formula for the generation of alpha and beta from the SE and mean.  
  
# The takeaway here is that my method for calculating SE is correct, as proven by Koen's excel file, and once I have the SE and the mean (mean is just the point estimate value I started with for this parameter) Andrew Briggs told me what to do to calculate the alpha and the beta, so I now can be confident in the manner in which I take the PSA for utility or probability when it comes to beta distributions.   
   
# mean<- 0.042178846  
# Maximum <- mean + 0.50\*mean  
# Maximum  
# Minimum <- mean - 0.40\*mean  
# Minimum  
#   
# se <- ((Maximum) - (mean)) / 2  
# se   
#   
# std.error <- se  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# beta  
#   
#   
#   
#   
# mean: 0.042178846  
# se: 0.01054471  
#   
# alpha =(((0.042178846)^2)\*(1-(0.042178846))/((0.01)^2)-(0.042178846))  
# beta =((1-(0.042178846))\*(((1-(0.042178846))\*(0.042178846))/((0.01)^2)-1))  
#  
# My alpha and beta:  
#  
# > alpha  
# [1] 15.28296  
# > beta  
# [1] 347.0541  
#  
# Koen's alpha and beta:  
# alpha = 14.961  
# beta = 363.614  
  
# mean<- 0.042178846  
  
# u\_ME = rbeta(10000, shape1 = 15.28296, shape2 = 347.0541)  
# mean(u\_ME)  
# [1] 0.04208022  
# u\_KOEN = rbeta(10000, shape1 = 14.961, shape2 = 363.614)  
# mean(u\_KOEN)  
# [1] 0.03940224  
# Which will both come to 0.04 if rounding to 2 decimal points  
  
#   
# mean= 0.042178846  
# se= 0.01  
#   
# newalpha =(((mean)^2)\*(1-(mean))/((se)^2)-(mean))  
# newbeta =((1-(mean))\*(((1-(mean))\*(mean))/((se)^2)-1))  
# > newalpha  
# [1] 16.99799  
# > newbeta  
# [1] 385.9999  
#   
#   
# manualalpha =(((0.042178846)^2)\*(1-(0.042178846))/((0.01)^2)-(0.042178846))  
# manualbeta =((1-(0.042178846))\*(((1-(0.042178846))\*(0.042178846))/((0.01)^2)-1))  
# > manualalpha  
# [1] 16.99799  
# > manualbeta  
# [1] 385.9999  
  
## RANGE:  
  
##### Calculating Beta values when you have a range around your point estimate in the literature (THIS CAN BE APPLIED TO PROBABILITY OR UTILITY):   
  
  
# Here are my formulas, and if you read further below you can see how I got to these formulas.  
  
# Here's the bottomline takeaway on the Beta formuala. My SE's are typically very close to the SE's that I recover from the published studies. I've seen first hand, that you need big changes to the SE to affect the mean you draw in the PSA. So, what I'll do is apply the briggse or altbriggse as appropriate, and then when I do the PSA I'll check the mean on the parameter drawn from the PSA draws to make sure the parameter is on average the same as the one we started with, I can do this by doing mean() whatever the parameter is, and that will show me the average value.  
  
# Brigs code to generate the SE for a parameter with a range perfectly centered around the mean (briggsse), and then repeated but for the situations where the range isnt PERFECTLY centered around the mean (altbriggsse), i.e., the min is further away from the mean than the max, or vice versa (even a little):  
  
# briggsse <- ((max)-(mean))/1.96  
# altbriggsse <- (max-min)/(2\*1.96)  
  
## Generating the alpha and beta:  
  
# mean <- Somevaluehere  
  
# std.error <- briggsse OR altbriggsse - as appropriate  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# beta  
#   
# u\_ME = rbeta(10000, shape1 = a, shape2 = b)  
# mean(u\_ME)  
  
# I could also create alpha and beta using Koen's code, which is a different way of calculating these vales than what Briggs showed me, but typically gives the same alpha and beta, so can be ignored:  
  
# newalpha =(((mean)^2)\*(1-(mean))/((se)^2)-(mean))  
# newbeta =((1-(mean))\*(((1-(mean))\*(mean))/((se)^2)-1))  
# newalpha  
# newbeta  
  
  
# Calculating a Beta distribution when you have a range around this in the literature you are reviewing:  
  
# All of the below applies to probabilities as well as utilities, as both are for beta distributions.  
  
# To determine the best way to do this, I open all the sources I mention just above which include a range, se, alpha, beta, etc., and then I apply all the methods I have and see which is best. I'll also be heavily guided by the suggestions Andy made, as he has being doing this work for 2 decades, if he says that a certain way is correct, I'll take him at his word. Lesley also made some suggestions.  
  
# Once you have an alpha and a beta, you can work your way back to a SE:  
  
  
# per: https://stackoverflow.com/questions/41189633/how-to-get-the-standard-deviation-from-the-fitted-distribution-in-scipy  
# As supported by this: https://www.real-statistics.com/binomial-and-related-distributions/beta-distribution/  
# and using this here: Jenks, Michelle, et al. "Tegaderm CHG IV securement dressing for central venous and arterial catheter insertion sites: a NICE medical technology guidance." Applied Health Economics and Health Policy 14.2 (2016): 135-149. https://link.springer.com/content/pdf/10.1007/s40258-015-0202-5.pdf I get 0.002096431 or 0.0021, so it looks like they used the mean as their se.  
  
  
# I demonstrate this in Table 1, row 22, of: Sharp, Linda, et al. "Cost-effectiveness of population-based screening for colorectal cancer: a comparison of guaiac-based faecal occult blood testing, faecal immunochemical testing and flexible sigmoidoscopy." British journal of cancer 106.5 (2012): 805-816. https://www.nature.com/articles/bjc2011580.pdf also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\bjc2011580.pdf  
  
# CTC sensitivity for CRC Basecase(85%) range(75 – 95%) Beta (Alpha = 50.00, Beta = 8.82)  
  
# a<- 50.00  
# b<- 8.82  
# Var <- a \* b / ( (a + b)^2 \* (a + b + 1) )  
# se <- sqrt(Var)  
# se  
# 0.04616056  
  
# Here's a way to get alpha and beta with just the variance and mean: https://devinincerti.com/2018/02/10/psa.html  
  
# If I wanted to turn my own SE into the variance, I can multiply it by itself, i.e. se\*se per: https://r-lang.com/how-to-calculate-square-of-all-values-in-r-vector/#:~:text=To%20calculate%20square%20in%20R,square%20of%20the%20input%20value.  
  
# mean <- 0.85  
  
# std.error <- se  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# 50.01124  
# beta  
# 8.825513  
  
  
# myse <- ((0.95) - (0.85)) / 2  
# myse  
# 0.05  
  
###### Below, I work through the differences that my SE formula lead to, and come to the realisation that an alternative formula informed by Briggs may be better: ######  
  
# The problem is, using this se changes my alpha and beta values, as follows:  
  
# std.error <- myse  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# 42.5  
# beta  
# 7.5  
  
# The differences between my results on their tables, and their results, may come down to rounding.  
  
# When I generate the SE from the values reported in the paper, I don't have as many decimal points for my SE, i.e., it rounds up to 0.05 from 0.04616056, as below:  
  
# myse <- ((0.95) - (0.85)) / 2  
# myse  
# 0.05  
  
# The problem is, using this rounded se changes my alpha and beta values, as follows:  
  
# std.error <- myse  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# 42.5  
# beta  
# 7.5  
  
# Which explains the differences in the alpha and beta I generate from the table.  
  
# What this means, is that I can generate my SE as before, that is   
  
# se <- ((Maximum) - (Mean)) / 2  
  
# Or se off the minimum if it's further away:  
  
# se <- ((Mean) - (Min)) / 2  
  
# But where there is a range, instead of just a point estimate value for utility, I use the upper end of the range for the maximum, or the lower end of the range for the minimum when generating the SE.  
  
# I've checked this for all the beta values in the Table. Only rows 1, 5, 20, 34 (out of the 17 beta rows) give results that are too different to be accounted for by rounding. To make sure I am 100% correct about this, I've emailed Lesley to ask how she got these distributions, as she is one of the co-authors on the paper. Lesley didnt do the analysis for this paper, and doesnt remember who did...  
  
# When the SE is only different by rounding, the alpha and beta that comes out of the formula after I've worked out the SE for their beta vars from the alpha and beta and plugged it into the formula matches the alpha and beta they started with, showing that the way in which I generate the SE is correct, because my se matches the se they use as below:  
  
# My standard error matches theirs (although involves rounding) and I get the same alpha and beta by including theirs or mine (or, at least I would without rounding) and gives me the same rbeta mean as they started out with. Subsequently, my thought process is this: I can take the approach of generating the SE from the upper range - mean or the mean - the lower range (depending on which is further away), and then plug this into my formula to get an appropriate alpha and beta and use this to make random draws for the mean in the rbeta. I won't produce an alpha and beta that matches what they present in the table, but the SE I recover from their alpha and beta, creates an unmatched version of the alpha and beta that matches the alpha and beta I create.   
  
# The fact that their standard error matches mine means that the way I am calculating SE in my own methods is correct.  
#   
# std.error <- se  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# beta  
#   
# u\_ME = rbeta(10000, shape1 = a, shape2 = b)  
# mean(u\_ME)  
  
#   
# std.error <- myse  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# beta  
#   
# u\_ME = rbeta(10000, shape1 = a, shape2 = b)  
# mean(u\_ME)  
  
  
# Then I create alpha and beta using Koen's code, which is a different way of calculating these vales than what Briggs showed me:  
  
# newalpha =(((mean)^2)\*(1-(mean))/((se)^2)-(mean))  
# newbeta =((1-(mean))\*(((1-(mean))\*(mean))/((se)^2)-1))  
# newalpha  
# [1] 50.01124  
# newbeta  
# [1] 8.825513  
  
# Gianluca Baio emailed me this code for calculating alpha and beta, which matches the above:  
  
# a <- m \* ((m \* (1 - m)/s^2) - 1)  
# b <- (1 - m) \* ((m \* (1 - m)/s^2) - 1)  
  
# According to Brett (McQueen, Robert <ROBERT.MCQUEEN@CUANSCHUTZ.EDU> in email): The beta distribution has two shape parameters calculated as alpha = ((mean^2)\*(1-mean)/(error)^2)-mean and beta = alpha\*((1-mean)/mean).  
  
# Spackman, D. E., & Veenstra, D. L. (2008). A cost-effectiveness analysis of currently approved treatments for HBeAg-positive chronic hepatitis B. Pharmacoeconomics, 26(11), 937-949. http://download.lww.com/wolterskluwer\_vitalstream\_com/permalink/pcz/a/00019053-920082611-00002.pdf Also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\00019053-920082611-00002.pdf  
  
  
  
# In the supplementary table to this paper, my SE is almost identical to the SE created in the paper, but because I am working with whole numbers, i.e., I only know 2 decimals of their code, say 0.03, whereas they would know lots more, say 0.031112234331, so my SE has less decimal points than there's, so of course, there will be slight variations when that SE is used for calculating alpha.plus.beta, which in turn has implications for alpha and beta calculated alone from this alpha.plus.beta, i.e., my calculation of these will be very slightly different because I am dividing by a very slightly different SE to create alpha plus beta. What I do see when I plug their longer SE into my formula I get the exact alpha and beta reported in the paper. This means that my formula to calculate alpha and beta is correct. Also, my SE is the exact same as theirs bar a few extra digits in theirs, for reasons described above, meaning that the way I calculate the SE is also perfect. Thus I can use the method described below for calculating SE from a parameter with a range, and thus the alpha and beta from this parameter, and thus the PSA distribution.  
  
  
# However, I do run into some cells where things are very different:  
  
# a <- 15.039072  
# b <- 736.9145  
# max <- 0.03  
# min <- 0.00  
# mean <-0.02  
  
# Var <- a\*b/((a+b)^2\*(a+b+1))  
# se<-sqrt(Var)  
# se  
# [1] 0.005102  
  
# myse <- ((max)-(mean))/2  
# #myse <- ((mean)-(min))/2  
# myse  
# [1] 0.005  
  
# std.error <- se  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# [1] 752  
  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# [1] 15.04  
# beta  
# [1] 736.9  
  
# u\_ME = rbeta(10000, shape1 = alpha, shape2 = beta)  
# mean(u\_ME)  
# [1] 0.01997  
  
# std.error <- myse  
# alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1 ## alpha + beta (ab)  
# alpha.plus.beta  
# [1] 783  
# alpha <- mean\*alpha.plus.beta ## alpha (a)  
# beta <- alpha\*(1-mean)/mean ## beta(b)  
# alpha  
# [1] 15.66  
# beta  
# [1] 767.3  
  
# u\_ME = rbeta(10000, shape1 = alpha, shape2 = beta)  
# mean(u\_ME)  
# [1] 0.02007  
  
  
# I create a vector with 1 million entries drawing randomly from between a minimum and a maximum value for the mean and max values featured in the paper:  
  
  
# possiblemeans <- runif(1000000, 0.012499999999999999, 0.0134999999999999999)  
# possiblemaxs <- runif(1000000, 0.022499999999999999, 0.0234999999999999999)  
  
# I create an SE that's made up of all these possible values  
  
# mypossibleses <- ((max)-(mean))/2  
# mypossibleses <- ((mean)-(min))/2  
# mypossibleses  
  
# The below tells me all the cells that have the included value :  
  
# grep(se, x1)  
  
# So, basically, what I am doing is taking all the possible numbers that could make up the max and the mean, putting them into my formula, and seeing if any permutation will give me the SE they report in their paper. If it does, this simply means that their SE and my SE differ because of the number of decimal places they have.  
  
# a <- 8.16870263  
# b <- 54.66747   
# max <- 0.23  
# min <- 0.07  
# mean <-0.13  
#   
# Var <- a\*b/((a+b)^2\*(a+b+1))  
# se<-sqrt(Var)  
# se  
# [1] 0.04209  
#   
# myse <- ((max)-(mean))/2  
# #myse <- ((mean)-(min))/2  
# myse  
# [1] 0.05  
  
  
# I create a vector with 1 million entries drawing randomly from between a minimum and a maximum valuefor the mean and max values featured in the paper:  
  
  
# possiblemeans <- runif(1000000, 0.12499999999999999, 0.134999999999999999)  
# possiblemaxs <- runif(1000000, 0.22499999999999999, 0.234999999999999999)  
# possiblemins <- runif(1000000, 0.06499999999999999, 0.074999999999999999)  
  
# I create an SE that's made up of all these possible values  
  
# mypossibleses <- ((possiblemaxs)-(possiblemeans))/2  
# mypossibleses <- ((possiblemeans)-(possiblemins))/2  
# mypossibleses  
  
# I create a data frame from this and then I control f to search for the se they created according to their alpha and beta:  
# se  
# mypossiblesesdf <- data.frame(mypossibleses)  
  
  
# I don't find it among my values.  
  
# This tells me that no matter what number of decimal places they had information on, I would never be able to generate the SE they generated from the values I am using. When I sort by the size of my data.frame the minimum value I have is 0.04501 and the maximum is 0.05499  
  
# I even tred dividing by all the t-quantiles:  
  
# tq <- c(1,1.376,1.963,3.078,6.31,12.71,31.82,63.66,106.1,159.2,0.816,1.061,1.386,1.886,2.92,4.303,6.965,9.925,12.85,15.76,0.765,0.978,1.25,1.638,2.353,3.182,4.541,5.841,6.994,8.053,0.741,0.941,1.19,1.533,2.132,2.776,3.747,4.604,5.321,5.951,0.727,0.92,1.156,1.476,2.015,2.571,3.365,4.032,4.57,5.03,0.718,0.906,1.134,1.44,1.943,2.447,3.143,3.707,4.152,4.524,0.711,0.896,1.119,1.415,1.895,2.365,2.998,3.499,3.887,4.207,0.706,0.889,1.108,1.397,1.86,2.306,2.896,3.355,3.705,3.991,0.703,0.883,1.1,1.383,1.833,2.262,2.821,3.25,3.573,3.835,0.7,0.879,1.093,1.372,1.812,2.228,2.764,3.169,3.472,3.716,0.697,0.876,1.088,1.363,1.796,2.201,2.718,3.106,3.393,3.624,0.695,0.873,1.083,1.356,1.782,2.179,2.681,3.055,3.33,3.55,0.694,0.87,1.079,1.35,1.771,2.16,2.65,3.012,3.278,3.489,0.692,0.868,1.076,1.345,1.761,2.145,2.624,2.977,3.234,3.438,0.691,0.866,1.074,1.341,1.753,2.131,2.602,2.947,3.197,3.395,0.69,0.865,1.071,1.337,1.746,2.12,2.583,2.921,3.165,3.358,0.689,0.863,1.069,1.333,1.74,2.11,2.567,2.898,3.138,3.326,0.688,0.862,1.067,1.33,1.734,2.101,2.552,2.878,3.113,3.298,0.688,0.861,1.066,1.328,1.729,2.093,2.539,2.861,3.092,3.273,0.687,0.86,1.064,1.325,1.725,2.086,2.528,2.845,3.073,3.251,0.686,0.859,1.063,1.323,1.721,2.08,2.518,2.831,3.056,3.231,0.686,0.858,1.061,1.321,1.717,2.074,2.508,2.819,3.041,3.214,0.685,0.858,1.06,1.319,1.714,2.069,2.5,2.807,3.027,3.198,0.685,0.857,1.059,1.318,1.711,2.064,2.492,2.797,3.014,3.183,0.684,0.856,1.058,1.316,1.708,2.06,2.485,2.787,3.003,3.17,0.684,0.856,1.058,1.315,1.706,2.056,2.479,2.779,2.992,3.158,0.684,0.855,1.057,1.314,1.703,2.052,2.473,2.771,2.982,3.147,0.683,0.855,1.056,1.313,1.701,2.048,2.467,2.763,2.973,3.136,0.683,0.854,1.055,1.311,1.699,2.045,2.462,2.756,2.965,3.127,0.683,0.854,1.055,1.31,1.697,2.042,2.457,2.75,2.957,3.118,0.682,0.853,1.054,1.309,1.696,2.04,2.453,2.744,2.95,3.109,0.682,0.853,1.054,1.309,1.694,2.037,2.449,2.738,2.943,3.102,0.682,0.853,1.053,1.308,1.692,2.035,2.445,2.733,2.937,3.094,0.682,0.852,1.052,1.307,1.691,2.032,2.441,2.728,2.931,3.088,0.682,0.852,1.052,1.306,1.69,2.03,2.438,2.724,2.926,3.081,0.681,0.851,1.05,1.303,1.684,2.021,2.423,2.704,2.902,3.055,0.679,0.849,1.047,1.299,1.676,2.009,2.403,2.678,2.87,3.018,0.679,0.848,1.045,1.296,1.671,2,2.39,2.66,2.849,2.994,0.677,0.845,1.041,1.289,1.658,1.98,2.358,2.617,2.798,2.935,0.674,0.842,1.036,1.282,1.645,1.96,2.326,2.576,2.748,2.878)  
  
# mypossibleses <- ((possiblemaxs)-(possiblemeans))/tq  
# mypossibleses  
# I create a data frame from this and then I control f to search for the se they created according to their alpha and beta:  
# se  
# mypossiblesesdf <- data.frame(mypossibleses, tq)  
  
# I ctrl f and searched 0.04209 and got a few hits, but when I went to look at these with the all the digits being displayed none of them matched exactly the long version of their SE, i.e. 0.0420918376330846  
  
# > print(mypossiblesesdf[748327,], digits = 22)  
  
# So, I'm definitely not able to create their SE from the formula I used  
  
# So, they definitely didnt create their SE from the same formula I used.  
  
  
  
# The solution:  
  
# [The below fact that there is a better formula may imply that the reason none of the million draws above match the SE recovered from the paper down to the decimal point is because I was using the wrong formula to calculte the SE, and maybe it would have matched given the correct formula from Briggs below]. Plus, even if they report 3 decimal points in the paper, you never know if they did their calculations with even more, say 5 decimal points, and as we saw above, even a few more decimal points can change the SE slightly, and thus the alpha plus beta and then the alpha and beta.  
  
# I take the code provided by Briggs, provided by Devin, and my own assumed code, and I apply it to the Packman paper. Basically, I recover the SE's for this paper by applying the SE recovery code to the alpha and beta they report, then I compare all the different methods I have to get the SE myself to see which one comes out right most of the time.  
  
  
# Here's my own code, the code provided by Briggs (in email) and the code reported by Devin Incerti (https://hesim-dev.github.io/hesim/articles/markov-cohort.html) when generating the SE:  
  
  
# b <-c(891.6005,15.41799,736.9145,1257.92,419.1321,48.37248,54.66747,759.661,1161.571,46.55078,371.0238,613.4266,1349.813,1007.815,150.3308,515.7984,1.572009,1.709663,1196.415,44.16013,432.4305,1143.101,478.3303,344.844,318.2237,839.0925,665.8365,668.3704,15334.68,2261.868,210.4119,295.9456,839.0925,295.9456,248.218,50.63905,51.89032,54.66747,265.7833)   
  
# a <-c(121.581888,51.6167467,15.039072,334.383756,62.6289387,12.09312,8.16870263,3.817392,254.979072,48.4508078,45.8568782,226.883808,449.9375,168.165112,22.4632179,113.224032,29.8681689,2.26629759,280.640564,9.04484592,22.7595,242.476063,28.3755279,38.316,14.64628,6.766875,12.20474,6.751216,15.35003,6.806023,6.731443,6.657262,6.766875,6.657262,6.625944,14.28281,15.76329,8.168703,6.814956)  
  
# max <-c(0.14,0.87,0.03,0.23,0.16,0.30,0.23,0.01,0.21,0.61,0.13,0.29,0.27,0.16,0.18,0.85,1.00,1.00,0.21,0.27,0.07,0.195,0.076,(13.0/100),(6.6/100),(1.6/100),(2.8/100),(2.0/100),(0.2/100),(0.6/100),(6.2/100),(4.4/100),(1.6/100),(4.4/100),(5.2/100),(32.0/100),(33.3/100),(23/100),(5.0/100))  
  
# min <-c(0.10,0.67,0.00,0.19,0.10,0.10,0.07,0.00,0.15,0.41,0.09,0.25,0.23,0.12,0.08,0.79,0.85,0.13,0.17,0.07,0.03,0.155,0.036,(7.0/100),(2.2/100),(0.4/100),(0.8/100),(0.5/100),(0.1/100),(0.015/100),(1.6/100),(1.1/100),(0.4/100),(1.1/100),(1.3/100),(12.0/100),(13.3/100),(6.5/100),(1.3/100))   
  
# mean <-c(0.12,0.77,0.02,0.21,0.13,0.20,0.13,0.005,0.18,0.51,0.11,0.27,0.25,0.14,0.13,0.18,0.95,0.57,0.19,0.17,0.05,0.175,0.056,(10.00/100),(4.4/100),(0.8/100),(1.8/100),(1.0/100),(0.1/100),(0.3/100),(3.1/100),(2.2/100),(0.8/100),(2.2/100),(2.6/100),(22.0/100),(23.3/100),(13.0/100),(2.5/100))  
  
# Recover the papers SE:  
  
# Just a quick aside that for the values in Table A4, I have to divide these by 100. They start off as just percents, so 10.0 (7.0, 13.0) to reflect 10% with a range of 7 to 13 percent. But, if I want them to be the same as the percentages I have throughout the model, i.e., 0 is 0% and 1 is 100%, and 0.5 is 50%, then I'll need to divide these by 100. This goes equally for any percentages I find in the literature reported as just 10% say, and want to include, they'll need to be divided by 100 to put them in the same units as the 0-1 percentages range I use throughout my model. And the fact that the alpha and beta calculated in this report exactly match the alpha and beta I calculate when I've made this /100) adjustment means that, regardless of the fact that the authors list the percentages as 10.0, 7.0 and 13.0 in Table A4, when they actually went to analyse these, they also applied the /100) adjustment.  
  
# Code to recover the SE's:  
  
  
# Var <- a\*b/((a+b)^2\*(a+b+1))  
# se<-sqrt(Var)  
  
# My own assumed SE code:  
# myse <- ((max)-(mean))/2  
# My own assumed SE code, using the min when the mean isnt centered in the range, so say the min is further from the mean than the max:  
# mysemin <- ((mean)-(min))/2  
  
# Devin's code to generate the SE, as described above:  
# devinse <- (max - min)/(2 \* qnorm(.975))  
  
# Brigs code to generate the SE, with the max - the mean, and then repeated but for the mean minus the min:  
# briggsse <- ((max)-(mean))/1.96  
# briggssemin <- ((mean)-(min))/1.96  
  
# Here are the results, for the paper:  
  
# myse: ------11-1--10-1--------00-  
# devinse: -------------11-1------  
# briggsse:11111110111111-111-111-  
# altbriggsse:11-1111111111  
# wrongbriggsse:1  
# logse:  
# logse2:  
# mysemin:  
# briggssemin:111111  
# logsemin2  
# tqse:  
  
# Obviously, briggsse is the most correct, most of the time (in this Table 1 represents an exact match, and - represents a close but not exact match, when something was wrong I either didnt include it, or included it as a 0 when it was really, really wrong). Only 12.82% of the time it was wrong. So, 87.18% of the time it was correct.  
  
# When the range isnt PERFECTLY centered around the mean, i.e., the min is further away from the mean than the max, it looks like you use altbriggsse to incorporate this wider range, than you would see taking max-mean, what's interesting is that mean-min isnt enough to replicate the results in the study, it has to be max-min (although the fact that mean-min is divided by /1.96 and max-min is divided by /2\*1.96 also contributes I'm sure). Devinse does this also, but I don't like it as much because the qnorm(.975) seems a bit less clear on what it's doing, even though > 2 \* qnorm(.975) [1] 3.919928 is more or less the exact same as > 2\*1.96 [1] 3.92. Nonetheless, Devins SE's are always slightly different to the actual SE when compared to briggsse, which has made me prefer briggsse.  
  
# So, the recipe is simple, use briggsse when the range is centered around the base case value, or point estimate, or mean, or however they describe it. But, When the range isnt PERFECTLY centered around the mean, i.e., the min is further away from the mean than the max, or vice versa (even a little), it looks like you use altbriggsse to incorporate this wider range. I've reviewed studies where Beta values refer to probabilities, and studies where Beta values refer to utilities, and the above still holds regardless.   
  
# altbriggsse <- (max-min)/(2\*1.96)  
  
  
# wrongbriggsse <- ((max)-(min))/2\*1.96  
  
# logse <- (log(max) - log(min)) / 3.92  
  
# 3.92 is just 1.96\*2  
  
# logse2 <- (log(max) - log(mean)) / (2\*1.96)  
# logsemin2 <- (log(mean) - log(min)) / (2\*1.96)  
  
# tq <- c(1,1.376,1.963,3.078,6.31,12.71,31.82,63.66,106.1,159.2,0.816,1.061,1.386,1.886,2.92,4.303,6.965,9.925,12.85,15.76,0.765,0.978,1.25,1.638,2.353,3.182,4.541,5.841,6.994,8.053,0.741,0.941,1.19,1.533,2.132,2.776,3.747,4.604,5.321,5.951,0.727,0.92,1.156,1.476,2.015,2.571,3.365,4.032,4.57,5.03,0.718,0.906,1.134,1.44,1.943,2.447,3.143,3.707,4.152,4.524,0.711,0.896,1.119,1.415,1.895,2.365,2.998,3.499,3.887,4.207,0.706,0.889,1.108,1.397,1.86,2.306,2.896,3.355,3.705,3.991,0.703,0.883,1.1,1.383,1.833,2.262,2.821,3.25,3.573,3.835,0.7,0.879,1.093,1.372,1.812,2.228,2.764,3.169,3.472,3.716,0.697,0.876,1.088,1.363,1.796,2.201,2.718,3.106,3.393,3.624,0.695,0.873,1.083,1.356,1.782,2.179,2.681,3.055,3.33,3.55,0.694,0.87,1.079,1.35,1.771,2.16,2.65,3.012,3.278,3.489,0.692,0.868,1.076,1.345,1.761,2.145,2.624,2.977,3.234,3.438,0.691,0.866,1.074,1.341,1.753,2.131,2.602,2.947,3.197,3.395,0.69,0.865,1.071,1.337,1.746,2.12,2.583,2.921,3.165,3.358,0.689,0.863,1.069,1.333,1.74,2.11,2.567,2.898,3.138,3.326,0.688,0.862,1.067,1.33,1.734,2.101,2.552,2.878,3.113,3.298,0.688,0.861,1.066,1.328,1.729,2.093,2.539,2.861,3.092,3.273,0.687,0.86,1.064,1.325,1.725,2.086,2.528,2.845,3.073,3.251,0.686,0.859,1.063,1.323,1.721,2.08,2.518,2.831,3.056,3.231,0.686,0.858,1.061,1.321,1.717,2.074,2.508,2.819,3.041,3.214,0.685,0.858,1.06,1.319,1.714,2.069,2.5,2.807,3.027,3.198,0.685,0.857,1.059,1.318,1.711,2.064,2.492,2.797,3.014,3.183,0.684,0.856,1.058,1.316,1.708,2.06,2.485,2.787,3.003,3.17,0.684,0.856,1.058,1.315,1.706,2.056,2.479,2.779,2.992,3.158,0.684,0.855,1.057,1.314,1.703,2.052,2.473,2.771,2.982,3.147,0.683,0.855,1.056,1.313,1.701,2.048,2.467,2.763,2.973,3.136,0.683,0.854,1.055,1.311,1.699,2.045,2.462,2.756,2.965,3.127,0.683,0.854,1.055,1.31,1.697,2.042,2.457,2.75,2.957,3.118,0.682,0.853,1.054,1.309,1.696,2.04,2.453,2.744,2.95,3.109,0.682,0.853,1.054,1.309,1.694,2.037,2.449,2.738,2.943,3.102,0.682,0.853,1.053,1.308,1.692,2.035,2.445,2.733,2.937,3.094,0.682,0.852,1.052,1.307,1.691,2.032,2.441,2.728,2.931,3.088,0.682,0.852,1.052,1.306,1.69,2.03,2.438,2.724,2.926,3.081,0.681,0.851,1.05,1.303,1.684,2.021,2.423,2.704,2.902,3.055,0.679,0.849,1.047,1.299,1.676,2.009,2.403,2.678,2.87,3.018,0.679,0.848,1.045,1.296,1.671,2,2.39,2.66,2.849,2.994,0.677,0.845,1.041,1.289,1.658,1.98,2.358,2.617,2.798,2.935,0.674,0.842,1.036,1.282,1.645,1.96,2.326,2.576,2.748,2.878)  
  
#tqse <- ((max)-(min))/tq  
# sprintf("%.100f",tqse <- ((max)-(min))/tq)  
# the above lets me choose how many decimal places to display in this piece of code (I've chosen 100 for some reason).  
  
# Below I review the results:  
  
# se  
# myse  
# devinse  
# briggsse  
# altbriggsse  
# wrongbriggsse  
# logse  
# logse2  
# mysemin  
# briggssemin  
# logsemin2  
# tqse  
  
  
# I apply this to other papers:  
  
# Picot, J., et al. "The clinical effectiveness and cost-effectiveness of bortezomib and thalidomide in combination regimens with an alkylating agent and a corticosteroid for the first-line treatment of multiple myeloma: a systematic review and economic evaluation." Health technology assessment (Winchester, England) 15.41 (2011): 1.  
  
# For this, briggsse is always right, while the other SE's are not.   
  
# Thanks for your email. I’m afraid the details are lost to me in the midst of time. Having a look at our model, we may have used different assumptions to the confidence intervals depending on what data were available, for example assuming a 10%, 20% increase of the mean for the CI. We then would have worked out the standard error from the confidence intervals as you describe in your email [briggse and altbriggse] and the alpha/beta are worked out from the se and mean. The formulas for alpha and beta are taken from Briggs et al. (Decision modelling for health economic evaluation, 2006).  
  
  
  
# Sharp, Linda, et al. "Cost-effectiveness of population-based screening for colorectal cancer: a comparison of guaiac-based faecal occult blood testing, faecal immunochemical testing and flexible sigmoidoscopy." British journal of cancer 106.5 (20book12): 805-816.  
  
# For this, I typically get approximations of the SE from the paper.  
  
# Thank you for getting in touch and it’s great that your work on the CRC model is progressing. Unfortunately, I don’t have an answer to your query. I was not actually involved in deriving these estimates and it is also quite a long time ago that I don’t remember who did the actual analysis - it would have been either Linda or Alan.  
  
# I think the approach that you are taking seems valid and reasonable so it may be better for you to use the simple estimate of +/-20%.  
  
# Sorry I am not able to provide more clarification.  
  
  
# Campbell, Jonathon R., et al. "Cost-effectiveness of latent tuberculosis infection screening before immigration to low-incidence countries." Emerging infectious diseases 25.4 (2019): 661.  
  
# For this, the SE is close, but typically a few percentage points off the recovered SE, so instead of 0.06 we might have 0.05, etc., which shouldnt really change the mean calculation much when doing the beta draws in the PSA.   
  
# Gao, L., Moodie, M., Mitchell, P. J., Churilov, L., Kleinig, T. J., Yassi, N., … & EXTEND-IA TNK Investigators. (2020). Cost-effectiveness of tenecteplase before thrombectomy for ischemic stroke. Stroke, 51(12), 3681-3689. https://www.ahajournals.org/doi/epdf/10.1161/STROKEAHA.120.029666  
  
# For this, the SE I generate is typically quite different to the SE they report.  
  
# # Lan Gao of "Cost-Effectiveness of Tenecteplase Before Thrombectomy for Ischemic Stroke" didnt use a range, instead he assumed SD=0.1\*mean (here the terms SE and SD are being used to refer to the same thing, Gianluca calss SE by SD also).   
  
  
#A cluster randomised trial, cost-effectiveness analysis and psychosocial evaluation of insulin pump therapy compared with multiple injections during flexible intensive insulin therapy for type 1 diabetes: the REPOSE Trial https://www.journalslibrary.nihr.ac.uk/hta/hta21200#/abstract  
  
# For this, the SE I recover from the alpha and beta matches their SE exactly, then I generate min and max as MaxMatch <- mean + 2\*se and MinMatch <- mean - 2\*se, and get the exact same results from myse <- ((max)-(mean))/2, meaning that they used my code to get their SE, briggs and devins SE code is almost identical in the SE it generates.  
  
  
  
# https://clas.ucdenver.edu/marcelo-perraillon/sites/default/files/attached-files/lecture\_13\_uncertainty.pdf also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\lecture\_13\_uncertainty.pdf  
  
# For this, the altbriggse code gets me SE's that are very similar to the SE's reported for these non-centered parameters.  
  
  
# Appendix F: Health economics report (2015 work undertaken by NICE Internal Clinical Guidelines) https://www.nice.org.uk/guidance/ng33/evidence/appendix-f-full-he-report-pdf-80851860763  
  
# For this: My SE's from Briggses are very close to the SE's recovered from the study, and when they are not, altbriggses are very close to those recovered from the study and the range is not symmetric around the point estimate, which is the whole reason for using altbriggses anyway.  
  
  
# Here's the takeaway on the Beta formuala. My SE's are typically very close to the SE's that I recover from the published studies. I've seen first hand, that you need big changes to the SE to affect the mean you draw in the PSA. So, what I'll do is apply the briggse or altbriggse as appropriate, and then when I do the PSA check the mean on the parameter drawn from the PSA draws to make sure the parameter is on average the same as the one we started with, I can do this by doing mean() whatever the parameter is, and that will show me the average value.  
  
# The following paper shows that my methodology above can be applied to papers with 2.5/97.5% percentiles and also shows that the code for myse is used in those rare situations where the mean, min and max are all the exact same number: A dynamic Bayesian Markov model for health economic evaluations of interventions in infectious disease https://sci-hub.ru/10.1186/s12874-018-0541-7 also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\haeussler2018.pdf  
  
# And speaking to Gianluca Baio directly in email, he tells me to calculate SE's in the same way in which I was doing this with myse.  
  
# # I think very similarly to what you did --- but say you think that the info is a range of .25 to .43, then you kind of figure out that the mean is sort of halfway through (~0.34). Then you assume that, approximately, the sd would be around 0.045 (because 0.34 +/- 2\*.045 gets you, approximately, in the range you have in the first place). [i.e. brings you to the min and the max you started with].  
  
# mean <-c(0.34)  
# min <-c(0.25)  
# max <-c(0.43)  
  
# myse <- ((max)-(mean))/2  
# myse <- ((0.43)-(0.34))/2  
  
# He says the se should be a number multiplied by two that brings you from the mean to the max or the mean to the min. So, ((0.43)-(0.34)) = 0.09, 0.09/2 = 0.045. The number 0.045 times 2 gives you 0.09, 0.34 + 0.09 gives you 0.43, i.e., the max.  
  
  
#  
  
  
  
  
  
  
  
  
  
  
  
  
############### Below are some incorrect solutions I came up with #############  
  
# Incorrect Solution 1:  
  
## My formulas and the need for a 95% CI:  
   
# As per: https://stats.stackexchange.com/questions/550293/how-to-calculate-standard-error-given-mean-and-confidence-interval-for-a-gamma-d/550892#550892  
   
   
# "I would like to note that, while those values in the table happen to correspond with ±2σ, the minimum and maximum values do not generally follow such simple formula with mean plus-minus some standard deviation.  
   
# In this case, the minimum and maximum values only correspond to the interval μ±2σ because the distribution seems to have been truncated at those values."  
   
   
# i.e., my formula will only work if the minimum and maximum values reported in a study are ±2 SE from the mean, i.e. + 2 SE from the mean for the maximum value, and -2 SE from the mean for the minimum value.  
   
# And that will be the case any time there is a 95% Confidence Interval:  
   
  
# [ "Since 95% of values fall within two standard deviations of the mean according to the 68-95-99.7 Rule, simply add and subtract two standard deviations from the mean in order to obtain the 95% confidence interval. Notice that with higher confidence levels the confidence interval gets large so there is less precision." https://www.westga.edu/academics/research/vrc/assets/docs/confidence\_intervals\_notes.pdf also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\gammadist\confidence\_intervals\_notes.pdf  
  
# A good image can be found as per: https://www.quora.com/What-is-meant-by-one-standard-deviation-away-from-the-mean also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\gammadist\www-quora-com-What-is-meant-by-.pdf]  
  
# So, if I'm taking my costs and a range about these costs from the literature, I just need to make sure the range is a 95% Confidence Interval, like that reported here: Tilson, L., Sharp, L., Usher, C., Walsh, C., Whyte, S., O’Ceilleachair, A., ... & Barry, M. (2012). Cost of care for colorectal cancer in Ireland: a health care payer perspective. The European journal of health economics, 13(4), 511-524. (https://www.researchgate.net/publication/51188456\_Cost\_of\_care\_for\_colorectal\_cancer\_in\_Ireland\_A\_health\_care\_payer\_perspective) reports a cost in Table 3 for Colorectal cancer with a 95% CI: € 48,835 (€40,548–€62,582)..  
  
# For "PK test cost MEAN: 400.00 MIN: 300.00 MAX: 500.00" from Table 4, row 3 page 222 in Goldstein, D. A., Chen, Q., Ayer, T., Howard, D. H., Lipscomb, J., Harvey, R. D., ... & Flowers, C. R. (2014). Cost effectiveness analysis of pharmacokinetically-guided 5-fluorouracil in FOLFOX chemotherapy for metastatic colorectal cancer. Clinical colorectal cancer, 13(4), 219-225, plugging the SE and mean into the CI = m ± t\*SE from Jochen above I get:  
   
# 400 + 2\*50 = 500  
   
# 400 - 2\*50 = 300  
   
# Which is the same confidence interval as I got in the Table 4 row 3, i.e. 300 and 500.  
   
# So, this might be a good way to check if my standard error is correct. Although this will only work if my confidence interval is centred on the mean, i.e. if I am the same distance from the mean to the max and to the min. If my confidence interval is NOT centred on the mean, I can calculate 2 standard errors, one between the lower interval and the mean, and one between the upper interval and the mean, and then apply these in CI = m ± t\*SE, knowing that:  
  
# CIupper = m + t\*SE  
  
# CIlower = m - t\*SE  
  
# So, I've figured out a simple test for whether the max/min reported is ±2σ, i.e., whether it is a 95% CI:  
# MaxMatch <- Mean + 2\*se   
# MinMatch <- Mean - 2\*se  
   
# I can then check if the max/min reported in a Table match the above, and if so I know that the max/min reported is ±2σ.  
   
# If not, it may be that they used the se off the minimum as it's further away, so try the below again, but with the minimums SE:  
  
# MaxMatch <- Mean + 2\*se   
# MinMatch <- Mean - 2\*se  
   
   
# I can then check if the max/min reported in a Table match the above, and if so I know that the max/min reported is ±2σ.  
   
   
# Even if it's not symmetric, this may still be a 95% confidence interval if the mean is at 2 of one of the standard errors from the min or the max, we can see this in action below:  
   
   
# > Maximum <- 27655.84  
# > Mean <- 11565.60  
# >   
# > se <- ((Maximum) - (Mean)) / 2  
# > se   
# [1] 8045.12  
# >   
# > MaxMatch <- Mean + 2\*se   
# > MinMatch <- Mean - 2\*se  
# > MaxMatch  
# [1] 27655.84  
# > MinMatch  
# [1] -4524.64  
   
# But still, for these values  
   
# FN: Value: 11,565.60 Minimum: 7092.16 Maximum: 27,655.84 Distribution: γ: (2.067, 5596.247)  
   
# that are not centered on the mean, this still came out the same as the Table using this se <- ((Maximum) - (Mean)) / 2 method. Because looking at the below, the upper interval (the maximum) is further from the mean at a difference of 16090.24, than the lower interval (the minimum) at a difference between it and the mean of 4473.44 (i.e. the maximum is nearly four times further away from the mean that the minimum).  
   
   
# > Maximum <- 27655.84  
# > Minimum <- 7092.16  
# > Mean <- 11565.60  
   
# > Maximum - Mean  
# [1] 16090.24  
   
# > Mean - Minimum  
# [1] 4473.44  
   
   
# And when we plug the se in for the further away max we get:  
   
   
# > a.cIntervention  
# [1] 2.066671  
# > b.cIntervention  
# [1] 5596.247  
   
# Just the same as the values for FN above.  
#   
  
# So, this kind of applies to situations where an interval is reported, when you might want to check if it's a 95% confidence interval, and if not, which interval is furthest from the mean.  
  
  
# So, I did this in the papers, by recovering the SE from the paper, and building a max and min that was 2 of these SE's from the mean as follows, and comparing this to the max and min reported in the paper:  
  
  
# Var <- a\*b/((a+b)^2\*(a+b+1))  
# se<-sqrt(Var)  
  
# MaxMatch <- mean + 2\*se   
# MinMatch <- mean - 2\*se  
  
# I found that the times where the interval wasnt like a 95% CI, i.e., the max and min were not 2 SE's either side of the mean away, was the times that my calculated SE and the papers calculated SE's didnt match. When they were close enough to 2 SE's either side, my formulas would calculate an SE that was a closer approximination to what was reported in the paper, but when they were further away than the 2 SE's either side of the mean, the SE's calculated from my formula would also be further away. The takeaway here is probably that, with a 95% CI, I can use my briggsse formula. Where the range isnt a 95% CI, but is pretty centered on the mean value I can use altbriggsse, when the range isnt a 95% CI, and isnt even slightly centered on the mean value then I should consider treating the mean as a point estimate and ignoring the range, to create my own SE which I know I can rely on.  
  
# Incorrect Solution 2:  
  
# Here's what I originally thought was happening in the Tilson paper, the formula to generate the se is se <- ((Maximum) - (Mean))/2 but the 2 is really a t-quantile with n-1 degrees of freedom [i.e., SE = (CIupper-m)/t], but I use 2 as a simplification because "For large n the quantile approaches 2.0 (well, 1.959964... to be more precise; but using 2.0 is good enough)." I think that the authors of this study actually looked up the t-quantile and divided by that each time, rather than dividing by 2, i.e. "The t-quantile can be looked up for the level of confidence when the total sample size (n) is known.... For instance, the t-quantile for 95% confidence, n=10 and k=2 is 2.3." per: researchgate.net/post/Formula\_for\_calculate\_Standard\_errorSE\_from\_Confidence\_IntervalCI There is also information supporting this here: https://stats.stackexchange.com/questions/550293/how-to-calculate-standard-error-given-mean-and-confidence-interval-for-a-gamma-d  
  
  
# I checked this by dividing by every value in the t-quantile here: https://homepage.cs.uiowa.edu/~jblang/probability.calculators/t.table.htm  
  
# tq <- c(1,1.376,1.963,3.078,6.31,12.71,31.82,63.66,106.1,159.2,0.816,1.061,1.386,1.886,2.92,4.303,6.965,9.925,12.85,15.76,0.765,0.978,1.25,1.638,2.353,3.182,4.541,5.841,6.994,8.053,0.741,0.941,1.19,1.533,2.132,2.776,3.747,4.604,5.321,5.951,0.727,0.92,1.156,1.476,2.015,2.571,3.365,4.032,4.57,5.03,0.718,0.906,1.134,1.44,1.943,2.447,3.143,3.707,4.152,4.524,0.711,0.896,1.119,1.415,1.895,2.365,2.998,3.499,3.887,4.207,0.706,0.889,1.108,1.397,1.86,2.306,2.896,3.355,3.705,3.991,0.703,0.883,1.1,1.383,1.833,2.262,2.821,3.25,3.573,3.835,0.7,0.879,1.093,1.372,1.812,2.228,2.764,3.169,3.472,3.716,0.697,0.876,1.088,1.363,1.796,2.201,2.718,3.106,3.393,3.624,0.695,0.873,1.083,1.356,1.782,2.179,2.681,3.055,3.33,3.55,0.694,0.87,1.079,1.35,1.771,2.16,2.65,3.012,3.278,3.489,0.692,0.868,1.076,1.345,1.761,2.145,2.624,2.977,3.234,3.438,0.691,0.866,1.074,1.341,1.753,2.131,2.602,2.947,3.197,3.395,0.69,0.865,1.071,1.337,1.746,2.12,2.583,2.921,3.165,3.358,0.689,0.863,1.069,1.333,1.74,2.11,2.567,2.898,3.138,3.326,0.688,0.862,1.067,1.33,1.734,2.101,2.552,2.878,3.113,3.298,0.688,0.861,1.066,1.328,1.729,2.093,2.539,2.861,3.092,3.273,0.687,0.86,1.064,1.325,1.725,2.086,2.528,2.845,3.073,3.251,0.686,0.859,1.063,1.323,1.721,2.08,2.518,2.831,3.056,3.231,0.686,0.858,1.061,1.321,1.717,2.074,2.508,2.819,3.041,3.214,0.685,0.858,1.06,1.319,1.714,2.069,2.5,2.807,3.027,3.198,0.685,0.857,1.059,1.318,1.711,2.064,2.492,2.797,3.014,3.183,0.684,0.856,1.058,1.316,1.708,2.06,2.485,2.787,3.003,3.17,0.684,0.856,1.058,1.315,1.706,2.056,2.479,2.779,2.992,3.158,0.684,0.855,1.057,1.314,1.703,2.052,2.473,2.771,2.982,3.147,0.683,0.855,1.056,1.313,1.701,2.048,2.467,2.763,2.973,3.136,0.683,0.854,1.055,1.311,1.699,2.045,2.462,2.756,2.965,3.127,0.683,0.854,1.055,1.31,1.697,2.042,2.457,2.75,2.957,3.118,0.682,0.853,1.054,1.309,1.696,2.04,2.453,2.744,2.95,3.109,0.682,0.853,1.054,1.309,1.694,2.037,2.449,2.738,2.943,3.102,0.682,0.853,1.053,1.308,1.692,2.035,2.445,2.733,2.937,3.094,0.682,0.852,1.052,1.307,1.691,2.032,2.441,2.728,2.931,3.088,0.682,0.852,1.052,1.306,1.69,2.03,2.438,2.724,2.926,3.081,0.681,0.851,1.05,1.303,1.684,2.021,2.423,2.704,2.902,3.055,0.679,0.849,1.047,1.299,1.676,2.009,2.403,2.678,2.87,3.018,0.679,0.848,1.045,1.296,1.671,2,2.39,2.66,2.849,2.994,0.677,0.845,1.041,1.289,1.658,1.98,2.358,2.617,2.798,2.935,0.674,0.842,1.036,1.282,1.645,1.96,2.326,2.576,2.748,2.878)  
  
# And when I checked if my SE now matched the SE I got from the alpha and beta in the paper, it still didnt match.  
  
# myse <- ((max)-(mean))/tq  
# myse  
  
  
  
# For Table 1 in the following, under utilities if I plug the SE and base case value into my formula for creating alpha and beta and then take randome draws for the mean from rbeta I get the same mean as reported as the base case value in the table. Indicating that the formula I use once I have the SE and mean to get the alpha and beta, and thus the new mean, is correct: Kristin, E., Endarti, D., Khoe, L. C., Taroeno-Hariadi, K. W., Trijayanti, C., Armansyah, A., & Sastroasmoro, S. (2021). Economic evaluation of adding bevacizumab to chemotherapy for metastatic colorectal cancer (mCRC) patients in Indonesia. Asian Pacific Journal of Cancer Prevention: APJCP, 22(6), 1921. also saved here: file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Evidence%20Synthesis/Economic%20Models/Kristin%20et%20al\_2021\_Economic%20Evaluation%20of%20Adding%20Bevacizumab%20to%20Chemotherapy%20for%20Metastatic.pdf  
  
  
# So, the whole problems comes down to the manner in which I am generating the SE from the mean and range.  
  
  
######################## SECTION ON INCORRECT SOLUTIONS ENDS ###################  
  
  
# Brigs code to generate the SE for a parameter with a range perfectly centered around the mean (briggsse), and then repeated but for the situations where the range isnt PERFECTLY centered around the mean (altbriggsse), i.e., the min is further away from the mean than the max, or vice versa (even a little):  
  
max <- 1.00,  
min <- 0.68,   
mean <- u\_F,  
  
# briggsse <- ((max)-(mean))/1.96,  
altbriggsse <- (max-min)/(2\*1.96),  
  
## Generating the alpha and beta:  
  
std.error <- altbriggsse, ## briggsse OR altbriggsse - as appropriate  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha.plus.beta,  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
u\_F = rbeta(n\_runs, shape1 = alpha, shape2 = beta),  
mean(u\_F),  
u\_F\_alpha <- alpha,  
u\_F\_beta <- beta,  
  
  
  
  
# Brigs code to generate the SE for a parameter with a range perfectly centered around the mean (briggsse), and then repeated but for the situations where the range isnt PERFECTLY centered around the mean (altbriggsse), i.e., the min is further away from the mean than the max, or vice versa (even a little):  
  
max <- 0.78,  
min <- 0.52,   
mean <- u\_P,  
  
briggsse <- ((max)-(mean))/1.96,  
# altbriggsse <- (max-min)/(2\*1.96),  
  
## Generating the alpha and beta:  
  
std.error <- briggsse, ## briggsse OR altbriggsse - as appropriate  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha.plus.beta,  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
u\_P = rbeta(n\_runs, shape1 = alpha, shape2 = beta),  
mean(u\_P),  
u\_P\_alpha <- alpha,  
u\_P\_beta <- beta,  
  
  
 u\_D = 0,  
# The utility of being dead doesnt vary, it always is the same utility to be a dead person, which is 0 because we don't have any quality of life, as we are no longer alive. I could have also just set u\_D = u\_D, because when you set something equal to itself in here, that also means it's constant.  
  
  
  
  
  
  
mean<- AE1\_DisUtil,  
Maximum <- Maximum\_AE1\_DisUtil,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
AE1\_DisUtil = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_u\_AE1 <- alpha,  
beta\_u\_AE1 <- beta,  
  
  
mean<- AE2\_DisUtil,  
Maximum <- Maximum\_AE2\_DisUtil,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
AE2\_DisUtil = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_u\_AE2 <- alpha,  
beta\_u\_AE2 <- beta,  
  
  
mean<- AE3\_DisUtil,  
Maximum <- Maximum\_AE3\_DisUtil,  
Maximum,  
se <- ((Maximum) - (mean)) / 2,  
se,   
std.error <- se,  
alpha.plus.beta <- mean\*(1-mean)/(std.error^2)-1, ## alpha + beta (ab)  
alpha <- mean\*alpha.plus.beta, ## alpha (a)  
beta <- alpha\*(1-mean)/mean, ## beta(b)  
alpha,  
beta,  
  
AE3\_DisUtil = rbeta(n\_runs, shape1 = alpha, shape2 = beta),   
  
alpha\_u\_AE3 <- alpha,  
beta\_u\_AE3 <- beta,  
  
  
  
  
  
  
# Discounting:  
   
# A uniform distribution is a probability distribution in which every value between an interval from a to b is equally likely to be chosen. https://www.statology.org/uniform-distribution-r/  
   
# "Generates random values that are evenly spread between min and max bounds" - https://docs.oracle.com/cd/E57185\_01/CYBUG/apcs03s03s01.html   
  
# Devin Incerti talks about the Uniform distribution here: "The uniform distribution is useful when there is little data available to estimate a parameter and determine its distribution. It is always preferable to place uncertainty on a parameter even when there is little evidence for it than to assume a fixed value (which gives a false sense of precision). Sampling from the uniform distribution is straightforward." https://devinincerti.com/2018/02/10/psa.html   
   
#Reading the literature, it seems like studies pick a lower bound and upper bound for the discount rate, such that the average (mean of the discount rate) will reach the point estimate they started with:   
   
 # A Trial-Based Assessment of the Cost-Utility of Bevacizumab and Chemotherapy versus Chemotherapy Alone for Advanced Non-Small Cell Lung Cancer https://sci-hub.ru/10.1016/j.jval.2011.04.004   
   
# file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Evidence%20Synthesis/Paper%20Materials%20and%20Methods/Hamdy%20Elsisi%20et%20al\_2019\_Cost-effectiveness%20of%20sorafenib%20versus%20best%20supportive%20care%20in%20advanced.pdf   
   
# Koen's excel file does this too: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Matching the model with the evidence Koen  
   
# as informed by: https://pubs.wsb.wisc.edu/academics/analytics-using-r-2019/uniform-continuous-version.html also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\3.5 Uniform (Continuous Version) \_ Analytics Using R.pdf  
  
  
# d\_c = 0.04/365,  
# d\_e = 0.04/365,  
# If I wanted constant values for the discount rates I would have set them as above.  
# Earlier I divided 0.04 by 365, so now I divide the upper value by 365 also.  
  
 d\_c = runif(n\_runs, min = 0, max = 0.08/365),  
 d\_e = runif(n\_runs, min = 0, max = 0.08/365),  
  
 n\_cycle = n\_cycle,  
 t\_cycle = t\_cycle  
)  
  
# Inspect the data.frame to see what it looks like (we'll see the first 6 observations):  
head(df\_PA\_input)

## coef\_weibull\_shape\_SoC coef\_weibull\_scale\_SoC coef\_TTD\_weibull\_shape\_SoC  
## 1 0.4295 5.606 0.3453  
## 2 0.3354 5.674 0.3215  
## 3 0.2912 5.562 0.4090  
## 4 0.3096 5.674 0.3678  
## 5 0.3319 5.646 0.3881  
## 6 0.2907 5.549 0.3973  
## coef\_TTD\_weibull\_scale\_SoC HR\_FP\_Exp HR\_PD\_Exp max....0.22 min....0.12  
## 1 6.633 0.7228 0.6747 0.22 0.12  
## 2 6.592 0.7449 0.7415 0.22 0.12  
## 3 6.688 0.6377 0.5860 0.22 0.12  
## 4 6.705 0.6745 0.5788 0.22 0.12  
## 5 6.589 0.8017 0.6369 0.22 0.12  
## 6 6.650 0.6615 0.8986 0.22 0.12  
## mean....P\_OSD\_SoC briggsse......max.....mean...1.96 std.error....briggsse  
## 1 0.17 0.02551 0.02551  
## 2 0.17 0.02551 0.02551  
## 3 0.17 0.02551 0.02551  
## 4 0.17 0.02551 0.02551  
## 5 0.17 0.02551 0.02551  
## 6 0.17 0.02551 0.02551  
## alpha.plus.beta....mean....1...mean...std.error.2....1 alpha.plus.beta  
## 1 215.8 215.8  
## 2 215.8 215.8  
## 3 215.8 215.8  
## 4 215.8 215.8  
## 5 215.8 215.8  
## 6 215.8 215.8  
## alpha....mean...alpha.plus.beta beta....alpha....1...mean..mean alpha beta  
## 1 36.69 179.1 36.69 179.1  
## 2 36.69 179.1 36.69 179.1  
## 3 36.69 179.1 36.69 179.1  
## 4 36.69 179.1 36.69 179.1  
## 5 36.69 179.1 36.69 179.1  
## 6 36.69 179.1 36.69 179.1  
## P\_OSD\_SoC mean.P\_OSD\_SoC. P\_OSD\_SoC\_alpha....alpha P\_OSD\_SoC\_beta....beta  
## 1 0.1590 0.17 36.69 179.1  
## 2 0.1708 0.17 36.69 179.1  
## 3 0.1621 0.17 36.69 179.1  
## 4 0.1675 0.17 36.69 179.1  
## 5 0.1472 0.17 36.69 179.1  
## 6 0.1411 0.17 36.69 179.1  
## max....0.22.1 min....0.12.1 mean....P\_OSD\_Exp  
## 1 0.22 0.12 0.17  
## 2 0.22 0.12 0.17  
## 3 0.22 0.12 0.17  
## 4 0.22 0.12 0.17  
## 5 0.22 0.12 0.17  
## 6 0.22 0.12 0.17  
## briggsse......max.....mean...1.96.1 std.error....briggsse.1  
## 1 0.02551 0.02551  
## 2 0.02551 0.02551  
## 3 0.02551 0.02551  
## 4 0.02551 0.02551  
## 5 0.02551 0.02551  
## 6 0.02551 0.02551  
## alpha.plus.beta....mean....1...mean...std.error.2....1.1 alpha.plus.beta.1  
## 1 215.8 215.8  
## 2 215.8 215.8  
## 3 215.8 215.8  
## 4 215.8 215.8  
## 5 215.8 215.8  
## 6 215.8 215.8  
## alpha....mean...alpha.plus.beta.1 beta....alpha....1...mean..mean.1 alpha.1  
## 1 36.69 179.1 36.69  
## 2 36.69 179.1 36.69  
## 3 36.69 179.1 36.69  
## 4 36.69 179.1 36.69  
## 5 36.69 179.1 36.69  
## 6 36.69 179.1 36.69  
## beta.1 P\_OSD\_Exp mean.P\_OSD\_Exp. P\_OSD\_Exp\_alpha....alpha  
## 1 179.1 0.1426 0.17 36.69  
## 2 179.1 0.1942 0.17 36.69  
## 3 179.1 0.1759 0.17 36.69  
## 4 179.1 0.1746 0.17 36.69  
## 5 179.1 0.1632 0.17 36.69  
## 6 179.1 0.1905 0.17 36.69  
## P\_OSD\_Exp\_beta....beta mean....p\_FA1\_STD Maximum....Maximum\_p\_FA1\_STD Maximum  
## 1 179.1 0.04 0.048 0.048  
## 2 179.1 0.04 0.048 0.048  
## 3 179.1 0.04 0.048 0.048  
## 4 179.1 0.04 0.048 0.048  
## 5 179.1 0.04 0.048 0.048  
## 6 179.1 0.04 0.048 0.048  
## se......Maximum.....mean...2 se std.error....se  
## 1 0.004 0.004 0.004  
## 2 0.004 0.004 0.004  
## 3 0.004 0.004 0.004  
## 4 0.004 0.004 0.004  
## 5 0.004 0.004 0.004  
## 6 0.004 0.004 0.004  
## alpha.plus.beta....mean....1...mean...std.error.2....1.2  
## 1 2399  
## 2 2399  
## 3 2399  
## 4 2399  
## 5 2399  
## 6 2399  
## alpha....mean...alpha.plus.beta.2 beta....alpha....1...mean..mean.2 alpha.2  
## 1 95.96 2303 95.96  
## 2 95.96 2303 95.96  
## 3 95.96 2303 95.96  
## 4 95.96 2303 95.96  
## 5 95.96 2303 95.96  
## 6 95.96 2303 95.96  
## beta.2 p\_FA1\_STD alpha\_p\_FA1\_STD....alpha beta\_p\_FA1\_STD....beta  
## 1 2303 0.03730 95.96 2303  
## 2 2303 0.03743 95.96 2303  
## 3 2303 0.04603 95.96 2303  
## 4 2303 0.03599 95.96 2303  
## 5 2303 0.04560 95.96 2303  
## 6 2303 0.04747 95.96 2303  
## mean....p\_FA2\_STD Maximum....Maximum\_p\_FA2\_STD Maximum.1  
## 1 0.31 0.372 0.372  
## 2 0.31 0.372 0.372  
## 3 0.31 0.372 0.372  
## 4 0.31 0.372 0.372  
## 5 0.31 0.372 0.372  
## 6 0.31 0.372 0.372  
## se......Maximum.....mean...2.1 se.1 std.error....se.1  
## 1 0.031 0.031 0.031  
## 2 0.031 0.031 0.031  
## 3 0.031 0.031 0.031  
## 4 0.031 0.031 0.031  
## 5 0.031 0.031 0.031  
## 6 0.031 0.031 0.031  
## alpha.plus.beta....mean....1...mean...std.error.2....1.3  
## 1 221.6  
## 2 221.6  
## 3 221.6  
## 4 221.6  
## 5 221.6  
## 6 221.6  
## alpha....mean...alpha.plus.beta.3 beta....alpha....1...mean..mean.3 alpha.3  
## 1 68.69 152.9 68.69  
## 2 68.69 152.9 68.69  
## 3 68.69 152.9 68.69  
## 4 68.69 152.9 68.69  
## 5 68.69 152.9 68.69  
## 6 68.69 152.9 68.69  
## beta.3 p\_FA2\_STD alpha\_p\_FA2\_STD....alpha beta\_p\_FA2\_STD....beta  
## 1 152.9 0.3273 68.69 152.9  
## 2 152.9 0.3542 68.69 152.9  
## 3 152.9 0.3877 68.69 152.9  
## 4 152.9 0.3460 68.69 152.9  
## 5 152.9 0.2884 68.69 152.9  
## 6 152.9 0.3610 68.69 152.9  
## mean....p\_FA3\_STD Maximum....Maximum\_p\_FA3\_STD Maximum.2  
## 1 0.31 0.372 0.372  
## 2 0.31 0.372 0.372  
## 3 0.31 0.372 0.372  
## 4 0.31 0.372 0.372  
## 5 0.31 0.372 0.372  
## 6 0.31 0.372 0.372  
## se......Maximum.....mean...2.2 se.2 std.error....se.2  
## 1 0.031 0.031 0.031  
## 2 0.031 0.031 0.031  
## 3 0.031 0.031 0.031  
## 4 0.031 0.031 0.031  
## 5 0.031 0.031 0.031  
## 6 0.031 0.031 0.031  
## alpha.plus.beta....mean....1...mean...std.error.2....1.4  
## 1 221.6  
## 2 221.6  
## 3 221.6  
## 4 221.6  
## 5 221.6  
## 6 221.6  
## alpha....mean...alpha.plus.beta.4 beta....alpha....1...mean..mean.4 alpha.4  
## 1 68.69 152.9 68.69  
## 2 68.69 152.9 68.69  
## 3 68.69 152.9 68.69  
## 4 68.69 152.9 68.69  
## 5 68.69 152.9 68.69  
## 6 68.69 152.9 68.69  
## beta.4 p\_FA3\_STD alpha\_p\_FA3\_STD....alpha beta\_p\_FA3\_STD....beta  
## 1 152.9 0.2878 68.69 152.9  
## 2 152.9 0.3633 68.69 152.9  
## 3 152.9 0.3070 68.69 152.9  
## 4 152.9 0.3490 68.69 152.9  
## 5 152.9 0.2790 68.69 152.9  
## 6 152.9 0.3080 68.69 152.9  
## mean....p\_FA1\_EXPR Maximum....Maximum\_p\_FA1\_EXPR Maximum.3  
## 1 0.07 0.084 0.084  
## 2 0.07 0.084 0.084  
## 3 0.07 0.084 0.084  
## 4 0.07 0.084 0.084  
## 5 0.07 0.084 0.084  
## 6 0.07 0.084 0.084  
## se......Maximum.....mean...2.3 se.3 std.error....se.3  
## 1 0.007 0.007 0.007  
## 2 0.007 0.007 0.007  
## 3 0.007 0.007 0.007  
## 4 0.007 0.007 0.007  
## 5 0.007 0.007 0.007  
## 6 0.007 0.007 0.007  
## alpha.plus.beta....mean....1...mean...std.error.2....1.5  
## 1 1328  
## 2 1328  
## 3 1328  
## 4 1328  
## 5 1328  
## 6 1328  
## alpha....mean...alpha.plus.beta.5 beta....alpha....1...mean..mean.5 alpha.5  
## 1 92.93 1235 92.93  
## 2 92.93 1235 92.93  
## 3 92.93 1235 92.93  
## 4 92.93 1235 92.93  
## 5 92.93 1235 92.93  
## 6 92.93 1235 92.93  
## beta.5 p\_FA1\_EXPR alpha\_p\_FA1\_EXPR....alpha beta\_p\_FA1\_EXPR....beta  
## 1 1235 0.07407 92.93 1235  
## 2 1235 0.06746 92.93 1235  
## 3 1235 0.07592 92.93 1235  
## 4 1235 0.06956 92.93 1235  
## 5 1235 0.06741 92.93 1235  
## 6 1235 0.05318 92.93 1235  
## mean....p\_FA2\_EXPR Maximum....Maximum\_p\_FA2\_EXPR Maximum.4  
## 1 0.11 0.132 0.132  
## 2 0.11 0.132 0.132  
## 3 0.11 0.132 0.132  
## 4 0.11 0.132 0.132  
## 5 0.11 0.132 0.132  
## 6 0.11 0.132 0.132  
## se......Maximum.....mean...2.4 se.4 std.error....se.4  
## 1 0.011 0.011 0.011  
## 2 0.011 0.011 0.011  
## 3 0.011 0.011 0.011  
## 4 0.011 0.011 0.011  
## 5 0.011 0.011 0.011  
## 6 0.011 0.011 0.011  
## alpha.plus.beta....mean....1...mean...std.error.2....1.6  
## 1 808.1  
## 2 808.1  
## 3 808.1  
## 4 808.1  
## 5 808.1  
## 6 808.1  
## alpha....mean...alpha.plus.beta.6 beta....alpha....1...mean..mean.6 alpha.6  
## 1 88.89 719.2 88.89  
## 2 88.89 719.2 88.89  
## 3 88.89 719.2 88.89  
## 4 88.89 719.2 88.89  
## 5 88.89 719.2 88.89  
## 6 88.89 719.2 88.89  
## beta.6 p\_FA2\_EXPR alpha\_p\_FA2\_EXPR....alpha beta\_p\_FA2\_EXPR....beta  
## 1 719.2 0.11395 88.89 719.2  
## 2 719.2 0.11087 88.89 719.2  
## 3 719.2 0.09936 88.89 719.2  
## 4 719.2 0.11393 88.89 719.2  
## 5 719.2 0.10623 88.89 719.2  
## 6 719.2 0.10671 88.89 719.2  
## mean....p\_FA3\_EXPR Maximum....Maximum\_p\_FA3\_EXPR Maximum.5  
## 1 0.07 0.084 0.084  
## 2 0.07 0.084 0.084  
## 3 0.07 0.084 0.084  
## 4 0.07 0.084 0.084  
## 5 0.07 0.084 0.084  
## 6 0.07 0.084 0.084  
## se......Maximum.....mean...2.5 se.5 std.error....se.5  
## 1 0.007 0.007 0.007  
## 2 0.007 0.007 0.007  
## 3 0.007 0.007 0.007  
## 4 0.007 0.007 0.007  
## 5 0.007 0.007 0.007  
## 6 0.007 0.007 0.007  
## alpha.plus.beta....mean....1...mean...std.error.2....1.7  
## 1 1328  
## 2 1328  
## 3 1328  
## 4 1328  
## 5 1328  
## 6 1328  
## alpha....mean...alpha.plus.beta.7 beta....alpha....1...mean..mean.7 alpha.7  
## 1 92.93 1235 92.93  
## 2 92.93 1235 92.93  
## 3 92.93 1235 92.93  
## 4 92.93 1235 92.93  
## 5 92.93 1235 92.93  
## 6 92.93 1235 92.93  
## beta.7 p\_FA3\_EXPR alpha\_p\_FA3\_EXPR....alpha beta\_p\_FA3\_EXPR....beta  
## 1 1235 0.07432 92.93 1235  
## 2 1235 0.07070 92.93 1235  
## 3 1235 0.06528 92.93 1235  
## 4 1235 0.06713 92.93 1235  
## 5 1235 0.05440 92.93 1235  
## 6 1235 0.07169 92.93 1235  
## Maximum....Maximum\_administration\_cost Mean....administration\_cost  
## 1 438 365  
## 2 438 365  
## 3 438 365  
## 4 438 365  
## 5 438 365  
## 6 438 365  
## se......Maximum.....Mean...2 se.6 mean....Mean mean mn.cIntervention....mean  
## 1 36.5 36.5 365 365 365  
## 2 36.5 36.5 365 365 365  
## 3 36.5 36.5 365 365 365  
## 4 36.5 36.5 365 365 365  
## 5 36.5 36.5 365 365 365  
## 6 36.5 36.5 365 365 365  
## se.cIntervention....se  
## 1 36.5  
## 2 36.5  
## 3 36.5  
## 4 36.5  
## 5 36.5  
## 6 36.5  
## a.cIntervention.....mn.cIntervention.se.cIntervention..2  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention.....se.cIntervention.2..mn.cIntervention a.cIntervention  
## 1 3.65 100  
## 2 3.65 100  
## 3 3.65 100  
## 4 3.65 100  
## 5 3.65 100  
## 6 3.65 100  
## b.cIntervention administration\_cost  
## 1 3.65 323.7  
## 2 3.65 400.6  
## 3 3.65 346.8  
## 4 3.65 370.2  
## 5 3.65 360.3  
## 6 3.65 322.9  
## a.cIntervention\_administration\_cost....a.cIntervention  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention\_administration\_cost....b.cIntervention  
## 1 3.65  
## 2 3.65  
## 3 3.65  
## 4 3.65  
## 5 3.65  
## 6 3.65  
## Maximum....Maximum\_c\_PFS\_Folfox Mean....c\_PFS\_Folfox  
## 1 369.4 307.8  
## 2 369.4 307.8  
## 3 369.4 307.8  
## 4 369.4 307.8  
## 5 369.4 307.8  
## 6 369.4 307.8  
## se......Maximum.....Mean...2.1 se.7 mean....Mean.1 mean.1  
## 1 30.78 30.78 307.8 307.8  
## 2 30.78 30.78 307.8 307.8  
## 3 30.78 30.78 307.8 307.8  
## 4 30.78 30.78 307.8 307.8  
## 5 30.78 30.78 307.8 307.8  
## 6 30.78 30.78 307.8 307.8  
## mn.cIntervention....mean.1 se.cIntervention....se.1  
## 1 307.8 30.78  
## 2 307.8 30.78  
## 3 307.8 30.78  
## 4 307.8 30.78  
## 5 307.8 30.78  
## 6 307.8 30.78  
## a.cIntervention.....mn.cIntervention.se.cIntervention..2.1  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention.....se.cIntervention.2..mn.cIntervention.1 a.cIntervention.1  
## 1 3.078 100  
## 2 3.078 100  
## 3 3.078 100  
## 4 3.078 100  
## 5 3.078 100  
## 6 3.078 100  
## b.cIntervention.1 c\_PFS\_Folfox  
## 1 3.078 345.8  
## 2 3.078 338.2  
## 3 3.078 273.3  
## 4 3.078 290.8  
## 5 3.078 308.7  
## 6 3.078 236.0  
## a.cIntervention\_c\_PFS\_Folfox....a.cIntervention  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention\_c\_PFS\_Folfox....b.cIntervention  
## 1 3.078  
## 2 3.078  
## 3 3.078  
## 4 3.078  
## 5 3.078  
## 6 3.078  
## Maximum....Maximum\_c\_PFS\_Bevacizumab Mean....c\_PFS\_Bevacizumab  
## 1 3096 2580  
## 2 3096 2580  
## 3 3096 2580  
## 4 3096 2580  
## 5 3096 2580  
## 6 3096 2580  
## se......Maximum.....Mean...2.2 se.8 mean....Mean.2 mean.2  
## 1 258 258 2580 2580  
## 2 258 258 2580 2580  
## 3 258 258 2580 2580  
## 4 258 258 2580 2580  
## 5 258 258 2580 2580  
## 6 258 258 2580 2580  
## mn.cIntervention....mean.2 se.cIntervention....se.2  
## 1 2580 258  
## 2 2580 258  
## 3 2580 258  
## 4 2580 258  
## 5 2580 258  
## 6 2580 258  
## a.cIntervention.....mn.cIntervention.se.cIntervention..2.2  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention.....se.cIntervention.2..mn.cIntervention.2 a.cIntervention.2  
## 1 25.8 100  
## 2 25.8 100  
## 3 25.8 100  
## 4 25.8 100  
## 5 25.8 100  
## 6 25.8 100  
## b.cIntervention.2 c\_PFS\_Bevacizumab  
## 1 25.8 2570  
## 2 25.8 2166  
## 3 25.8 2533  
## 4 25.8 2578  
## 5 25.8 2711  
## 6 25.8 2686  
## a.cIntervention\_c\_PFS\_Bevacizumab....a.cIntervention  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention\_c\_PFS\_Bevacizumab....b.cIntervention  
## 1 25.8  
## 2 25.8  
## 3 25.8  
## 4 25.8  
## 5 25.8  
## 6 25.8  
## Maximum....Maximum\_c\_OS\_Folfiri Mean....c\_OS\_Folfiri  
## 1 391.2 326  
## 2 391.2 326  
## 3 391.2 326  
## 4 391.2 326  
## 5 391.2 326  
## 6 391.2 326  
## se......Maximum.....Mean...2.3 se.9 mean....Mean.3 mean.3  
## 1 32.6 32.6 326 326  
## 2 32.6 32.6 326 326  
## 3 32.6 32.6 326 326  
## 4 32.6 32.6 326 326  
## 5 32.6 32.6 326 326  
## 6 32.6 32.6 326 326  
## mn.cIntervention....mean.3 se.cIntervention....se.3  
## 1 326 32.6  
## 2 326 32.6  
## 3 326 32.6  
## 4 326 32.6  
## 5 326 32.6  
## 6 326 32.6  
## a.cIntervention.....mn.cIntervention.se.cIntervention..2.3  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention.....se.cIntervention.2..mn.cIntervention.3 a.cIntervention.3  
## 1 3.26 100  
## 2 3.26 100  
## 3 3.26 100  
## 4 3.26 100  
## 5 3.26 100  
## 6 3.26 100  
## b.cIntervention.3 c\_OS\_Folfiri  
## 1 3.26 306.9  
## 2 3.26 260.8  
## 3 3.26 361.6  
## 4 3.26 365.3  
## 5 3.26 319.6  
## 6 3.26 264.3  
## a.cIntervention\_c\_OS\_Folfiri....a.cIntervention  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention\_c\_OS\_Folfiri....b.cIntervention c\_D Maximum....Maximum\_c\_AE1  
## 1 3.26 0 3403  
## 2 3.26 0 3403  
## 3 3.26 0 3403  
## 4 3.26 0 3403  
## 5 3.26 0 3403  
## 6 3.26 0 3403  
## Mean....c\_AE1 se......Maximum.....Mean...2.4 se.10 mean....Mean.4 mean.4  
## 1 2836 283.6 283.6 2836 2836  
## 2 2836 283.6 283.6 2836 2836  
## 3 2836 283.6 283.6 2836 2836  
## 4 2836 283.6 283.6 2836 2836  
## 5 2836 283.6 283.6 2836 2836  
## 6 2836 283.6 283.6 2836 2836  
## mn.cIntervention....mean.4 se.cIntervention....se.4  
## 1 2836 283.6  
## 2 2836 283.6  
## 3 2836 283.6  
## 4 2836 283.6  
## 5 2836 283.6  
## 6 2836 283.6  
## a.cIntervention.....mn.cIntervention.se.cIntervention..2.4  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention.....se.cIntervention.2..mn.cIntervention.4 a.cIntervention.4  
## 1 28.36 100  
## 2 28.36 100  
## 3 28.36 100  
## 4 28.36 100  
## 5 28.36 100  
## 6 28.36 100  
## b.cIntervention.4 c\_AE1 a.cIntervention\_c\_AE1....a.cIntervention  
## 1 28.36 2786 100  
## 2 28.36 3244 100  
## 3 28.36 2828 100  
## 4 28.36 2492 100  
## 5 28.36 2371 100  
## 6 28.36 2744 100  
## b.cIntervention\_c\_AE1....b.cIntervention Maximum....Maximum\_c\_AE2  
## 1 28.36 1751  
## 2 28.36 1751  
## 3 28.36 1751  
## 4 28.36 1751  
## 5 28.36 1751  
## 6 28.36 1751  
## Mean....c\_AE2 se......Maximum.....Mean...2.5 se.11 mean....Mean.5 mean.5  
## 1 1459 145.9 145.9 1459 1459  
## 2 1459 145.9 145.9 1459 1459  
## 3 1459 145.9 145.9 1459 1459  
## 4 1459 145.9 145.9 1459 1459  
## 5 1459 145.9 145.9 1459 1459  
## 6 1459 145.9 145.9 1459 1459  
## mn.cIntervention....mean.5 se.cIntervention....se.5  
## 1 1459 145.9  
## 2 1459 145.9  
## 3 1459 145.9  
## 4 1459 145.9  
## 5 1459 145.9  
## 6 1459 145.9  
## a.cIntervention.....mn.cIntervention.se.cIntervention..2.5  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention.....se.cIntervention.2..mn.cIntervention.5 a.cIntervention.5  
## 1 14.59 100  
## 2 14.59 100  
## 3 14.59 100  
## 4 14.59 100  
## 5 14.59 100  
## 6 14.59 100  
## b.cIntervention.5 c\_AE2 a.cIntervention\_c\_AE2....a.cIntervention  
## 1 14.59 1455 100  
## 2 14.59 1716 100  
## 3 14.59 1609 100  
## 4 14.59 1288 100  
## 5 14.59 1665 100  
## 6 14.59 1483 100  
## b.cIntervention\_c\_AE2....b.cIntervention Maximum....Maximum\_c\_AE3  
## 1 14.59 490.8  
## 2 14.59 490.8  
## 3 14.59 490.8  
## 4 14.59 490.8  
## 5 14.59 490.8  
## 6 14.59 490.8  
## Mean....c\_AE3 se......Maximum.....Mean...2.6 se.12 mean....Mean.6 mean.6  
## 1 409 40.9 40.9 409 409  
## 2 409 40.9 40.9 409 409  
## 3 409 40.9 40.9 409 409  
## 4 409 40.9 40.9 409 409  
## 5 409 40.9 40.9 409 409  
## 6 409 40.9 40.9 409 409  
## mn.cIntervention....mean.6 se.cIntervention....se.6  
## 1 409 40.9  
## 2 409 40.9  
## 3 409 40.9  
## 4 409 40.9  
## 5 409 40.9  
## 6 409 40.9  
## a.cIntervention.....mn.cIntervention.se.cIntervention..2.6  
## 1 100  
## 2 100  
## 3 100  
## 4 100  
## 5 100  
## 6 100  
## b.cIntervention.....se.cIntervention.2..mn.cIntervention.6 a.cIntervention.6  
## 1 4.09 100  
## 2 4.09 100  
## 3 4.09 100  
## 4 4.09 100  
## 5 4.09 100  
## 6 4.09 100  
## b.cIntervention.6 c\_AE3 a.cIntervention\_c\_AE3....a.cIntervention  
## 1 4.09 375.9 100  
## 2 4.09 432.0 100  
## 3 4.09 329.0 100  
## 4 4.09 429.5 100  
## 5 4.09 380.2 100  
## 6 4.09 381.5 100  
## b.cIntervention\_c\_AE3....b.cIntervention max....1 min....0.68 mean....u\_F  
## 1 4.09 1 0.68 0.85  
## 2 4.09 1 0.68 0.85  
## 3 4.09 1 0.68 0.85  
## 4 4.09 1 0.68 0.85  
## 5 4.09 1 0.68 0.85  
## 6 4.09 1 0.68 0.85  
## altbriggsse.....max...min...2...1.96. std.error....altbriggsse  
## 1 0.08163 0.08163  
## 2 0.08163 0.08163  
## 3 0.08163 0.08163  
## 4 0.08163 0.08163  
## 5 0.08163 0.08163  
## 6 0.08163 0.08163  
## alpha.plus.beta....mean....1...mean...std.error.2....1.8 alpha.plus.beta.2  
## 1 18.13 18.13  
## 2 18.13 18.13  
## 3 18.13 18.13  
## 4 18.13 18.13  
## 5 18.13 18.13  
## 6 18.13 18.13  
## alpha....mean...alpha.plus.beta.8 beta....alpha....1...mean..mean.8 alpha.8  
## 1 15.41 2.72 15.41  
## 2 15.41 2.72 15.41  
## 3 15.41 2.72 15.41  
## 4 15.41 2.72 15.41  
## 5 15.41 2.72 15.41  
## 6 15.41 2.72 15.41  
## beta.8 u\_F mean.u\_F. u\_F\_alpha....alpha u\_F\_beta....beta max....0.78  
## 1 2.72 0.8669 0.85 15.41 2.72 0.78  
## 2 2.72 0.9104 0.85 15.41 2.72 0.78  
## 3 2.72 0.7911 0.85 15.41 2.72 0.78  
## 4 2.72 0.7779 0.85 15.41 2.72 0.78  
## 5 2.72 0.9294 0.85 15.41 2.72 0.78  
## 6 2.72 0.8624 0.85 15.41 2.72 0.78  
## min....0.52 mean....u\_P briggsse......max.....mean...1.96.2  
## 1 0.52 0.65 0.06633  
## 2 0.52 0.65 0.06633  
## 3 0.52 0.65 0.06633  
## 4 0.52 0.65 0.06633  
## 5 0.52 0.65 0.06633  
## 6 0.52 0.65 0.06633  
## std.error....briggsse.2  
## 1 0.06633  
## 2 0.06633  
## 3 0.06633  
## 4 0.06633  
## 5 0.06633  
## 6 0.06633  
## alpha.plus.beta....mean....1...mean...std.error.2....1.9 alpha.plus.beta.3  
## 1 50.71 50.71  
## 2 50.71 50.71  
## 3 50.71 50.71  
## 4 50.71 50.71  
## 5 50.71 50.71  
## 6 50.71 50.71  
## alpha....mean...alpha.plus.beta.9 beta....alpha....1...mean..mean.9 alpha.9  
## 1 32.96 17.75 32.96  
## 2 32.96 17.75 32.96  
## 3 32.96 17.75 32.96  
## 4 32.96 17.75 32.96  
## 5 32.96 17.75 32.96  
## 6 32.96 17.75 32.96  
## beta.9 u\_P mean.u\_P. u\_P\_alpha....alpha u\_P\_beta....beta u\_D  
## 1 17.75 0.6770 0.65 32.96 17.75 0  
## 2 17.75 0.7057 0.65 32.96 17.75 0  
## 3 17.75 0.6445 0.65 32.96 17.75 0  
## 4 17.75 0.7649 0.65 32.96 17.75 0  
## 5 17.75 0.7018 0.65 32.96 17.75 0  
## 6 17.75 0.6397 0.65 32.96 17.75 0  
## mean....AE1\_DisUtil Maximum....Maximum\_AE1\_DisUtil Maximum.6  
## 1 0.45 0.54 0.54  
## 2 0.45 0.54 0.54  
## 3 0.45 0.54 0.54  
## 4 0.45 0.54 0.54  
## 5 0.45 0.54 0.54  
## 6 0.45 0.54 0.54  
## se......Maximum.....mean...2.6 se.13 std.error....se.6  
## 1 0.045 0.045 0.045  
## 2 0.045 0.045 0.045  
## 3 0.045 0.045 0.045  
## 4 0.045 0.045 0.045  
## 5 0.045 0.045 0.045  
## 6 0.045 0.045 0.045  
## alpha.plus.beta....mean....1...mean...std.error.2....1.10  
## 1 121.2  
## 2 121.2  
## 3 121.2  
## 4 121.2  
## 5 121.2  
## 6 121.2  
## alpha....mean...alpha.plus.beta.10 beta....alpha....1...mean..mean.10  
## 1 54.55 66.67  
## 2 54.55 66.67  
## 3 54.55 66.67  
## 4 54.55 66.67  
## 5 54.55 66.67  
## 6 54.55 66.67  
## alpha.10 beta.10 AE1\_DisUtil alpha\_u\_AE1....alpha beta\_u\_AE1....beta  
## 1 54.55 66.67 0.3976 54.55 66.67  
## 2 54.55 66.67 0.4539 54.55 66.67  
## 3 54.55 66.67 0.4483 54.55 66.67  
## 4 54.55 66.67 0.4742 54.55 66.67  
## 5 54.55 66.67 0.3977 54.55 66.67  
## 6 54.55 66.67 0.5031 54.55 66.67  
## mean....AE2\_DisUtil Maximum....Maximum\_AE2\_DisUtil Maximum.7  
## 1 0.19 0.228 0.228  
## 2 0.19 0.228 0.228  
## 3 0.19 0.228 0.228  
## 4 0.19 0.228 0.228  
## 5 0.19 0.228 0.228  
## 6 0.19 0.228 0.228  
## se......Maximum.....mean...2.7 se.14 std.error....se.7  
## 1 0.019 0.019 0.019  
## 2 0.019 0.019 0.019  
## 3 0.019 0.019 0.019  
## 4 0.019 0.019 0.019  
## 5 0.019 0.019 0.019  
## 6 0.019 0.019 0.019  
## alpha.plus.beta....mean....1...mean...std.error.2....1.11  
## 1 425.3  
## 2 425.3  
## 3 425.3  
## 4 425.3  
## 5 425.3  
## 6 425.3  
## alpha....mean...alpha.plus.beta.11 beta....alpha....1...mean..mean.11  
## 1 80.81 344.5  
## 2 80.81 344.5  
## 3 80.81 344.5  
## 4 80.81 344.5  
## 5 80.81 344.5  
## 6 80.81 344.5  
## alpha.11 beta.11 AE2\_DisUtil alpha\_u\_AE2....alpha beta\_u\_AE2....beta  
## 1 80.81 344.5 0.2194 80.81 344.5  
## 2 80.81 344.5 0.2090 80.81 344.5  
## 3 80.81 344.5 0.1865 80.81 344.5  
## 4 80.81 344.5 0.1809 80.81 344.5  
## 5 80.81 344.5 0.1911 80.81 344.5  
## 6 80.81 344.5 0.2070 80.81 344.5  
## mean....AE3\_DisUtil Maximum....Maximum\_AE3\_DisUtil Maximum.8  
## 1 0.36 0.432 0.432  
## 2 0.36 0.432 0.432  
## 3 0.36 0.432 0.432  
## 4 0.36 0.432 0.432  
## 5 0.36 0.432 0.432  
## 6 0.36 0.432 0.432  
## se......Maximum.....mean...2.8 se.15 std.error....se.8  
## 1 0.036 0.036 0.036  
## 2 0.036 0.036 0.036  
## 3 0.036 0.036 0.036  
## 4 0.036 0.036 0.036  
## 5 0.036 0.036 0.036  
## 6 0.036 0.036 0.036  
## alpha.plus.beta....mean....1...mean...std.error.2....1.12  
## 1 176.8  
## 2 176.8  
## 3 176.8  
## 4 176.8  
## 5 176.8  
## 6 176.8  
## alpha....mean...alpha.plus.beta.12 beta....alpha....1...mean..mean.12  
## 1 63.64 113.1  
## 2 63.64 113.1  
## 3 63.64 113.1  
## 4 63.64 113.1  
## 5 63.64 113.1  
## 6 63.64 113.1  
## alpha.12 beta.12 AE3\_DisUtil alpha\_u\_AE3....alpha beta\_u\_AE3....beta  
## 1 63.64 113.1 0.4002 63.64 113.1  
## 2 63.64 113.1 0.3636 63.64 113.1  
## 3 63.64 113.1 0.3401 63.64 113.1  
## 4 63.64 113.1 0.3618 63.64 113.1  
## 5 63.64 113.1 0.3304 63.64 113.1  
## 6 63.64 113.1 0.4056 63.64 113.1  
## d\_c d\_e n\_cycle t\_cycle  
## 1 0.00020978 0.00008106 143 14  
## 2 0.00008086 0.00012278 143 14  
## 3 0.00015620 0.00005348 143 14  
## 4 0.00009401 0.00014531 143 14  
## 5 0.00005925 0.00010241 143 14  
## 6 0.00008537 0.00002045 143 14

# It's a dataframe made up of the 10,000 values I asked be made at the start of this code chunk.  
  
# If I wanted to save the dataframe, I would do this as follows:  
  
#save(df\_PA\_input, file = "df\_PA\_input.rda")  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
# Histogram of parameters  
  
  
# I believe they make a histogram of parameters to show the distribution of parameters, and thus indicate which distribution they should be using in their sensitivity analysis. Just like slide 18 of C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\A212ProbabilisticDistributions-210604-154.pdf which is also a HISTOGRAM OF RANDOM DRAWS FOR RELATIVE RISK and the related notes in C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\notes.txt from [SLIDE 18] onward.  
  
# i.e., "And that's exactly what Andrews done here showing the histogram of random draws from the relative risk parameter, and you see that, in fact, in this case, with quite a lot of information [I guess the nine trials with a total of 2,541 patients pooled], we wouldn't have been that far off if we'd assumed a normally distributed relative risk [i.e. our distribution looks very similar to a normal distribution]. But you can probably detect a very slight skew to that distribution, which reveals it actually as a log normally distributed relative risk."  
  
  
ggplot(melt(df\_PA\_input, variable.name = "Parameter"), aes(x = value)) +  
 facet\_wrap(~Parameter, scales = "free") +  
 geom\_histogram(aes(y = ..density..)) +  
 theme\_bw(base\_size = 16) +   
 theme(axis.text = element\_text(size=8))



I need to fix the “# Histogram of parameters” section above, the below gives advice on doing this:

<https://stackoverflow.com/questions/68416435/rcpp-package-doesnt-include-rcpp-precious-remove>

<https://www.mail-archive.com/rcpp-devel@lists.r-forge.r-project.org/msg10226.html>

<https://statisticsglobe.com/warning-cannot-remove-prior-installation-in-r>

The above approach worked, I manually deleted the package, installed darthtools from github, chose option 4, i.e. replace rccp and then I was done.

## 09.1 Conduct probabilistic sensitivity analysis

To propagate uncertainty throughout the model, we assigned distributions to any parameters not estimated with certainty. We consider whether values within a plausible range for input variables alter the conclusions of cost-effectiveness drawn from the original analysis.

To evaluate the robustness of the model, we apply deterministic methods (point estimates with an appropriate range) and probabilistic methods (parameterized distributions) to conduct a sensitivity analysis of our results. C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf

# Running the probabilistic analysis :  
  
# First we need to create data.frames to store the output from the PSA:  
  
df\_c <- df\_e <- data.frame(  
 SoC = rep(NA, n\_runs),  
 Exp = rep(NA, n\_runs)  
)  
  
# As you'll see, first we make blank (repeat NA for the number of runs of the simulationn\_runs) data.frames for costs (df\_c) and effectiveness (df\_e) for SoC and the experimental treatment:  
  
head(df\_c)

## SoC Exp  
## 1 NA NA  
## 2 NA NA  
## 3 NA NA  
## 4 NA NA  
## 5 NA NA  
## 6 NA NA

# > head(df\_c)  
# SoC Exp  
# 1 NA NA  
# 2 NA NA  
# 3 NA NA  
# 4 NA NA  
# 5 NA NA  
# 6 NA NA  
  
head(df\_e)

## SoC Exp  
## 1 NA NA  
## 2 NA NA  
## 3 NA NA  
## 4 NA NA  
## 5 NA NA  
## 6 NA NA

# > head(df\_e)  
# SoC Exp  
# 1 NA NA  
# 2 NA NA  
# 3 NA NA  
# 4 NA NA  
# 5 NA NA  
# 6 NA NA  
  
# We run the Markov model for each set of parameter values from the PSA input dataset (a set of parameters here is the row of parameter values in the PSA data.frame from whatever run number of the nruns in the PSA dataset we are on, i.e. if we were doing this for run number 10 out of 10,000, we would be on row 10 in the data.frame (as each row refers to a run) and we would use all the values for costs and effect that appear in this row when running the Markov model. So, we would be using all the random draws for that PSA run in the Markov model).  
  
# - I read a note indicating that this loop can be run in parallel to decrease the runtime, something to consider in the future if it takes a very long time...  
  
for(i\_run in 1:n\_runs){  
  
 # Evaluate the model and store the outcomes  
 l\_out\_temp <- oncologySemiMarkov(l\_params\_all = df\_PA\_input[i\_run, ], n\_wtp = 45000)  
# The above is basically saying, apply the oncologySemiMarkov\_function, where l\_params\_all in the function (i.e. the list of parameters the function is to be applied to) is equal to the parameters from the PSA data.frame for the run we are in, and the willingness to pay threshold is 45,000.  
 df\_c[i\_run, ] <- l\_out\_temp$Cost  
 df\_e[i\_run, ] <- l\_out\_temp$Effect  
# The above says, for the costs and effectiveness data.frames, store the value of costs and effects for each run that we are on in a row that reflects that run (remembering that [ROW,COLUMN], we are putting each cost and effect in a row that relates to the number of the run we are on, so if we are on run 1, i\_run = 1 and we store the costs and effects for that run 1 in row 1. Remembering that our oncologySemiMarkov makes the "Cost" and "Effect" parameters at the end of the function, after applying the Markov cost-effectiveness model to all our input data for the cost-effectiveness model (so, things like, cost, utility and probability), so we are just pulling the ouputted Cost and Effect values calculated from a Markov cost-effectiveness model applied to our input data after the input data has been randomly drawn from a PSA (per the PSA input dataframe)).   
  
   
 # While we're doing this, we might like to display the progress of the simulation:  
 if(i\_run/(n\_runs/10) == round(i\_run/(n\_runs/10), 0)) { # We've chosen to display progress every 10%  
 cat('\r', paste(i\_run/n\_runs \* 100, "% done", sep = " "))  
 }  
}

## 10 % done 20 % done 30 % done 40 % done 50 % done 60 % done 70 % done 80 % done 90 % done 100 % done

## 09.2 Create PSA object for dampack

# Dampack has a number of functions to summarise and visualise the results of a probabilistic sensitivity analysis. However, the data needs to be in a specific structure before we can use those functions.   
  
# The 'dampack' package contains multiple useful functions that summarize and visualize the results of a  
# probabilitic analysis. To use those functions, the data has to be in a particular structure.  
l\_PA <- make\_psa\_obj(cost = df\_c,   
 effectiveness = df\_e,   
 parameters = df\_PA\_input,   
 strategies = c("SoC", "Exp"))  
  
# So, basically we make a psa object for Dampack where df\_c is the dataframe of costs from the Markov model being applied to the PSA data, and df\_e is the data.frame of effectiveness from the Markov model being applied to the PSA dataset, the parameters that are included are those from the PSA analysis, which we fed into df\_PA\_input above (df\_PA\_input<-) and the two strategies are SoC and Exp.

## 09.2.1 Save PSA objects

# If we wanted to save the PSA objects once they have been created above, we could do this as follows (v\_names\_strats is just another way of including the strategies part from above):  
# save(df\_PA\_input, df\_c, df\_e, v\_names\_strats, n\_str, l\_PA,  
# file = "markov\_3state\_PSA\_dataset.RData")

## 09.3.1 Conduct CEA with probabilistic output

In the PSA we computed model outcomes (total discounted costs and QALYs) for the experimental strategy and standard of care for each sampled set of parameter values. Results were remarkably similar to the deterministic analysis, per the probabilistic league table below [or else I could say, eith ICER of X, incremental costs of Y and incremental effectiveness of Y, i.e. describe the league table, but describe it in words, i.e., a mean incremental cost of xx,xxx (95% CI xx,xxx - xx,xxx) with a mean incremental benefit of x.xx QALYs (95% CI x.xx - x.xx), resulted in a mean ICER of xxx,xxx per QALY (95% CI xxx,xxx - xxx,xxx).], demonstrating the robustness of the model.

The values estimated using means from the PSA (costs, QALYs and ICER) reflect those estimated in the deterministic analysis.

:

# First we take the mean outcome estimates, that is, we summarise the expected costs and effects for each strategy from the PSA:  
(df\_out\_ce\_PA <- summary(l\_PA))

## Strategy meanCost meanEffect  
## 1 SoC 24778 0.3372  
## 2 Exp 78992 0.6493

# Calculate incremental cost-effectiveness ratios (ICERs)  
  
# Then we calculate the ICERs from this df\_out\_ce\_PA (summary must be a dampack function doing things under the hood to create a selectable ($) meanCost, meanEffect and Strategy parameter in df\_out\_ce\_PA):  
(df\_cea\_PA <- calculate\_icers(cost = df\_out\_ce\_PA$meanCost,   
 effect = df\_out\_ce\_PA$meanEffect,  
 strategies = df\_out\_ce\_PA$Strategy))

## Strategy Cost Effect Inc\_Cost Inc\_Effect ICER Status  
## 1 SoC 24778 0.3372 NA NA NA ND  
## 2 Exp 78992 0.6493 54214 0.3121 173707 ND

# We can view the ICER results from the PSA here:  
df\_cea\_PA

## Strategy Cost Effect Inc\_Cost Inc\_Effect ICER Status  
## 1 SoC 24778 0.3372 NA NA NA ND  
## 2 Exp 78992 0.6493 54214 0.3121 173707 ND

# If we wanted to save the CEA (or leauge) table with ICERs, we would do this as follows:  
# As .RData  
# save(df\_cea\_pa,   
# file = "markov\_3state\_probabilistic\_CEA\_results.RData")  
# As .csv  
# write.csv(df\_cea\_pa,   
# file = "markov\_3state\_probabilistic\_CEA\_results.csv")  
  
  
## CEA table in proper format ---- per: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\analysis\cSTM\_time\_dep\_simulation.r  
table\_cea\_PA <- format\_table\_cea(df\_cea\_PA) # Function included in "R/Functions.R"; depends on the `scales` package  
table\_cea\_PA

## Strategy Costs ($) QALYs Incremental Costs ($) Incremental QALYs  
## 1 SoC 24,778 0.34 <NA> NA  
## 2 Exp 78,992 0.65 54,214 0.31  
## ICER ($/QALY) Status  
## 1 <NA> ND  
## 2 173,707 ND

## 09.3.2 Cost-Effectiveness Scatter plot

This indicates that what I am displaying here is not the cost-effectiveness plane, but a cost-effectiveness scatter plot: <https://yhec.co.uk/glossary/cost-effectiveness-plane/>

Devin does the traditional cost-effectiveness plane (but DARTH may also offer this in Dampack, and I should dig in there first if I want to draw a more traditional cost-effectiveness plane) <https://hesim-dev.github.io/hesim/articles/cea.html>

Figure 3 here shows the same scatterplt as I make, but parts of it cross the axis, I suspect that my scatterplot would cross the axis if necessary and thus reflect the above planes: Cost-effectiveness of capecitabine and bevacizumab maintenance treatment after first-line induction treatment in metastatic colorectal cancer sci-hub.ru/10.1016/j.ejca.2017.01.019 Actually. That would never happen. The axis needs an origin point. Typically ,the origin point is the SoC’s cost and effect, than we can say, look, does Exp fall below this cost (so our points for cost are lower on the graph, below the origin point) or above (so our costs are above this point on the graph, higher up above the origin point) and how does are utility look, higher than the SoC, so further to the right where effectiveness is the numberline along the bottom on the x-axis, or worse than SoC, so to the left of the SoC origin point, so a lower utility as we get further down the numberline and away from the SoC origin point for utility. It’s easiest to understand this while looking at Figure 3 below, where C is the origin point, because it A) reflects what I am saying above, and B) Shows that even the SoC maybe can be made of random draws too (although I’m not super sure on that, because if that value keeps changing then the comparative value of the ExP would be different always, so probably we need to still take the SoC value for the origin point we started with from the deterministic CEA so that we have something to compare our probabilistic ExP values to, but then we show the variability in SoC values around this from the PSA draws for SoC). Realistically, it’s probably best to produce a scatterplot as provided by dampack, and if a reviewer then wants the cost-effectiveness plane, to look into how this can be done in dampack.

This is supported by Figure 3 of: Probabilistic Analysis of Cost-Effectiveness Models: Choosing between Treatment Strategies for Gastroesophageal Reflux Disease Andrew H. Briggs, DPhil, Ron Goeree, MA, Gord Blackhouse, MBA, Bernie J. O’Brien, PhD med.mcgill.ca/epidemiology/courses/EPIB654/Summer2010/EF/example%20PPI.pdf

The cost-effectiveness plane illustrates the distribution of costs and effects for SoC and Exp.

Incremental costs and QALYs with their respective ICERs.

These values are illustrated in a cost-effectiveness plane of incremental costs in the y axis and incremental QALYs in the x axis.

We randomly sampled from each specified parameter distribution in the probabilistic analysis, calculating the expected costs and effects of that combination of parameter values. This reflects a single replication of model results, in total, 10,000 replications were completed to determine the possible distribution of the resulting ICERs for both SoC and the experimental treatment. In Figure X we plot the results of this probabilistic sensitivity analysis for these 10,000 replications from the model.

In Figure X we plot a probabilistic sensitivity analysis for 10,000 samples.

[**Legend Underneath:** Fig. X. Probabilistic Sensitivity Analysis (n = 10,000). Each dot represents an incremental cost-effectiveness ratio, ellipse represent the 95% confidence incidence.]

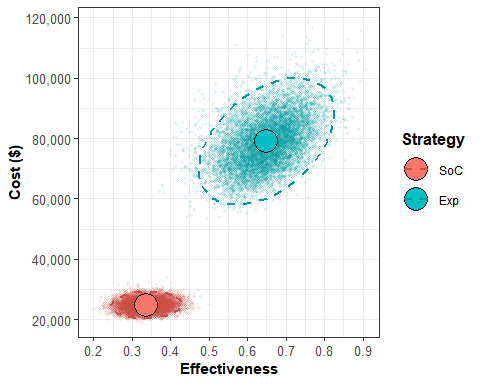
Well explained here: *An Introductory Tutorial on Cohort State-Transition Models in R Using a Cost-Effectiveness Analysis Example* [*https://arxiv.org/pdf/2001.07824.pdf*](https://arxiv.org/pdf/2001.07824.pdf)

3.2. Cost-effectiveness analysis The cost-effectiveness outcomes obtained from the DT-STM and the DES model are presented in the incremental cost-effectiveness planes of Fig. 4. The incremental effectiveness estimates, including their 95% confidence intervals (CI), for CAP-B maintenance therapy compared to the observation strategy are 0.21 (CI: 0.015; 0.430) and 0.18 (CI: 0.006; 0.374) QALYs, and the incremental costs are €35,536 (CI: 19,945; 54,629) and €30,053, (CI: 17,047; 46,132) for the DT-STM and DES model, respectively. The mean ICERs are €172,443 and €168,383 per QALY gained for the DT-STM and DES model, respectively. The PSA for both models only demonstrated a small difference in the amount of uncertainty surrounding the mean ICER point-estimates (Fig. 4). This is illustrated by the magnitude of the 95%-confidence ellipses surrounding these estimates, being slightly smaller for the DES model. However, as both mean ICER point-estimates and corresponding confidence ellipses are located rather similarly compared to the willingness- to-pay (WTP) threshold, the CEACs for both models are similar (Fig. 5). per: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Matching the model with the evidence Koen

Also see here for write up:

Sharp, L., Tilson, L., Whyte, S., O’Ceilleachair, A., Walsh, C., Usher, C., … & Comber, H. (2012). Cost-effectiveness of population-based screening for colorectal cancer: a comparison of guaiac-based faecal occult blood testing, faecal immunochemical testing and flexible sigmoidoscopy. British journal of cancer, 106(5), 805-816. <https://www.nature.com/articles/bjc2011580.pdf>

# Incremental cost-effectiveness plane  
plot(l\_PA)



#plot(l\_PA, xlim = c(9.5, 22.5))  
# In the DARTH package R code they use the additional bits above, I can see how they look in my own model, and if I'd like to use them.

## 09.4.2 Cost-effectiveness acceptability curves (CEACs) and frontier (CEAF)

Cost-effectiveness acceptability curves describe the probability of cost-effectiveness (i.e. that the ICER is below the WTP value) for each treatment option.

We calculated the probability the novel strategy would be cost-effective under several willingness to pay thresholds and illustrated this with a cost-effectiveness acceptability curve.

We derived cost-effectiveness acceptability curves (CEACs) from the probabilistic sensitivity analysis. The results are shown in CEACs in Figure X. This describes the probability that the addition of Bevacizumab was cost-effective across increasing willingness-to-pay (WTP) values. Results demonstrate a 100% probability that this strategy was more cost-effective when the threshold was X, and a xx.xx% probability of cost-effectiveness when the threshold was Y. These results indicate that base case analysis produces robust results.

Borrowing from: A Cost-Effectiveness Analysis of Currently Approved Treatments for HBeAg-Positive Chronic Hepatitis B <https://sci-hub.ru/10.2165/00019053-200826110-00006> I can say:

The results of the probabilistic sensitivity analysis indicate that treatment X is preferred at cost-effectiveness thresholds less than approximately $X,000 per QALY, and treatment Y had the highest probability of being optimal above this willingness to pay threshold.

Understand this plot here:

<https://cran.r-project.org/web/packages/dampack/vignettes/basic_cea.html>

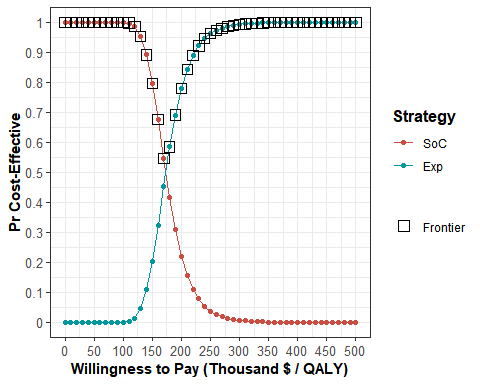
Also saved here:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\sensitivity analysis help files\Basic Cost Effectiveness Analysis.pdf

# Cost-effectiveness acceptability curve (CEAC):  
  
# We first generate a vector of willingness to pay values (thresholds) to define for which values the CEAC is made:  
v\_wtp <- seq(0, 500000, by = 10000)  
# This basically gives you a sequence of willingness to pay thresholds from 0 all the way up to 500,000 euro, increasing by 10,000 euro each time, so start with 0 euro, go to 10,000 euro, go to 20,000 euro, and so on until you hit 500,000. You can chose any max value you like and any increment value you like, for example, in the DARTH material they use the following: v\_wtp <- seq(0, 45000, by = 1000)  
CEAC\_obj <- ceac(wtp = v\_wtp, psa = l\_PA)  
  
  
# The below provides details on the regions of highest probability of cost-effectiveness for each strategy  
summary(CEAC\_obj)

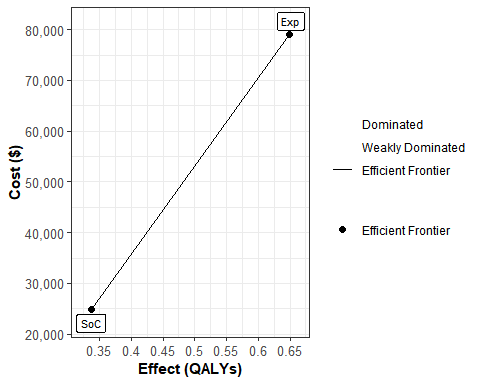
## range\_min range\_max cost\_eff\_strat  
## 1 0 180000 SoC  
## 2 180000 500000 Exp

# CEAC and cost-effectiveness acceptability frontier (CEAF) plot  
plot(CEAC\_obj)



## 09.4.3 Plot cost-effectiveness frontier

plot(df\_cea\_PA)



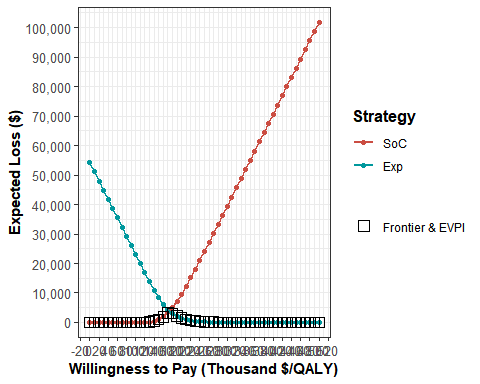
## 09.4.4 Expected Loss Curves (ELCs)

The expected loss is the the quantification of the foregone benefits when choosing a suboptimal strategy given current evidence.

elc\_obj <- calc\_exp\_loss(wtp = v\_wtp, psa = l\_PA)  
elc\_obj

## WTP Strategy Expected\_Loss On\_Frontier  
## 1 0 SoC 0.00000 TRUE  
## 2 0 Exp 54213.58762 FALSE  
## 3 10000 SoC 0.00000 TRUE  
## 4 10000 Exp 51092.60155 FALSE  
## 5 20000 SoC 0.00000 TRUE  
## 6 20000 Exp 47971.61548 FALSE  
## 7 30000 SoC 0.00000 TRUE  
## 8 30000 Exp 44850.62941 FALSE  
## 9 40000 SoC 0.00000 TRUE  
## 10 40000 Exp 41729.64334 FALSE  
## 11 50000 SoC 0.00000 TRUE  
## 12 50000 Exp 38608.65727 FALSE  
## 13 60000 SoC 0.00000 TRUE  
## 14 60000 Exp 35487.67120 FALSE  
## 15 70000 SoC 0.00000 TRUE  
## 16 70000 Exp 32366.68513 FALSE  
## 17 80000 SoC 0.00000 TRUE  
## 18 80000 Exp 29245.69906 FALSE  
## 19 90000 SoC 0.00000 TRUE  
## 20 90000 Exp 26124.71299 FALSE  
## 21 100000 SoC 0.87788 TRUE  
## 22 100000 Exp 23004.60479 FALSE  
## 23 110000 SoC 6.14922 TRUE  
## 24 110000 Exp 19888.89007 FALSE  
## 25 120000 SoC 35.26657 TRUE  
## 26 120000 Exp 16797.02135 FALSE  
## 27 130000 SoC 142.46876 TRUE  
## 28 130000 Exp 13783.23747 FALSE  
## 29 140000 SoC 425.09773 TRUE  
## 30 140000 Exp 10944.88037 FALSE  
## 31 150000 SoC 982.83865 TRUE  
## 32 150000 Exp 8381.63521 FALSE  
## 33 160000 SoC 1921.65735 TRUE  
## 34 160000 Exp 6199.46785 FALSE  
## 35 170000 SoC 3280.78465 TRUE  
## 36 170000 Exp 4437.60907 FALSE  
## 37 180000 SoC 5065.32491 FALSE  
## 38 180000 Exp 3101.16327 TRUE  
## 39 190000 SoC 7215.68422 FALSE  
## 40 190000 Exp 2130.53650 TRUE  
## 41 200000 SoC 9653.69122 FALSE  
## 42 200000 Exp 1447.55744 TRUE  
## 43 210000 SoC 12305.02698 FALSE  
## 44 210000 Exp 977.90712 TRUE  
## 45 220000 SoC 15113.07111 FALSE  
## 46 220000 Exp 664.96518 TRUE  
## 47 230000 SoC 18020.92577 FALSE  
## 48 230000 Exp 451.83377 TRUE  
## 49 240000 SoC 20999.23306 FALSE  
## 50 240000 Exp 309.15499 TRUE  
## 51 250000 SoC 24026.12898 FALSE  
## 52 250000 Exp 215.06483 TRUE  
## 53 260000 SoC 27082.84217 FALSE  
## 54 260000 Exp 150.79196 TRUE  
## 55 270000 SoC 30158.80465 FALSE  
## 56 270000 Exp 105.76836 TRUE  
## 57 280000 SoC 33248.14694 FALSE  
## 58 280000 Exp 74.12458 TRUE  
## 59 290000 SoC 36347.08950 FALSE  
## 60 290000 Exp 52.08107 TRUE  
## 61 300000 SoC 39453.01671 FALSE  
## 62 300000 Exp 37.02221 TRUE  
## 63 310000 SoC 42562.69884 FALSE  
## 64 310000 Exp 25.71828 TRUE  
## 65 320000 SoC 45676.05888 FALSE  
## 66 320000 Exp 18.09224 TRUE  
## 67 330000 SoC 48791.08321 FALSE  
## 68 330000 Exp 12.13050 TRUE  
## 69 340000 SoC 51908.31203 FALSE  
## 70 340000 Exp 8.37325 TRUE  
## 71 350000 SoC 55027.11902 FALSE  
## 72 350000 Exp 6.19417 TRUE  
## 73 360000 SoC 58146.55276 FALSE  
## 74 360000 Exp 4.64184 TRUE  
## 75 370000 SoC 61266.34524 FALSE  
## 76 370000 Exp 3.44825 TRUE  
## 77 380000 SoC 64386.45332 FALSE  
## 78 380000 Exp 2.57026 TRUE  
## 79 390000 SoC 67506.82971 FALSE  
## 80 390000 Exp 1.96057 TRUE  
## 81 400000 SoC 70627.30003 FALSE  
## 82 400000 Exp 1.44483 TRUE  
## 83 410000 SoC 73747.90538 FALSE  
## 84 410000 Exp 1.06410 TRUE  
## 85 420000 SoC 76868.52100 FALSE  
## 86 420000 Exp 0.69365 TRUE  
## 87 430000 SoC 79989.17889 FALSE  
## 88 430000 Exp 0.36548 TRUE  
## 89 440000 SoC 83110.05492 FALSE  
## 90 440000 Exp 0.25544 TRUE  
## 91 450000 SoC 86230.93460 FALSE  
## 92 450000 Exp 0.14905 TRUE  
## 93 460000 SoC 89351.81428 FALSE  
## 94 460000 Exp 0.04265 TRUE  
## 95 470000 SoC 92472.75770 FALSE  
## 96 470000 Exp 0.00000 TRUE  
## 97 480000 SoC 95593.74377 FALSE  
## 98 480000 Exp 0.00000 TRUE  
## 99 490000 SoC 98714.72984 FALSE  
## 100 490000 Exp 0.00000 TRUE  
## 101 500000 SoC 101835.71591 FALSE  
## 102 500000 Exp 0.00000 TRUE

# ELC plot  
plot(elc\_obj, log\_y = FALSE)



## 09.4.4 Expected value of perfect information (EVPI)

A value-of-information analysis estimates the expected value of perfect information (EVPI), that is,

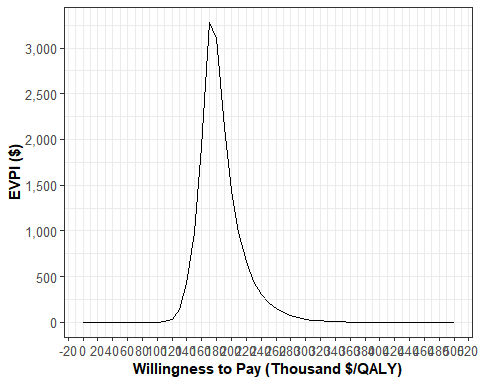
Value of information is discussed in the York course and below:

C:- Royal College of Surgeons in IrelandResources\_Exclusive\_Decision Modeling for Public Health\_DARTH

<https://cran.r-project.org/web/packages/dampack/vignettes/voi.html>

There’s also a paper on this here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Decision Modelling - Advanced Course\A2\_Making Models Probabilistic\A2.1.2 Distributions for parameters\Betadist utility\EVSI.doc

# Expected value of perfect information (EVPI)  
EVPI\_obj <- calc\_evpi(wtp = v\_wtp, psa = l\_PA)  
# EVPI plot  
plot(EVPI\_obj, effect\_units = "QALY")



## 09.4.5 Alternative plotting approaches:

You can find alternative plotting approaches here:

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\analysis\cSTM\_time\_dep\_simulation.R

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\manuscript\cSTM\_Tutorial\_TimeDep.Rmd

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\manuscript\cSTM\_Tutorial\_TimeDep.R

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\manuscript\Appendix\_code\_walkthrough\_Markov\_Tutorial\_Part2.Rmd

There may be information more generally in the R files found in: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\R-HTA in LMICs Intermediate R-HTA Modelling Tutorial\September-Workshop-main\September-Workshop-main\

## 08.3.3 Table Describing Parameters

To build a Table describing parameters, you click on the Table button at the top, and design it in there. Then to include the parameter values automatically as they are updated you fill in the parameter name you want, highlight it in the cell and click </> from above to put it in a code block. You do the same for gamma, etc., put this in code blocks also. <https://rpruim.github.io/s341/S19/from-class/MathinRmd.html#:~:text=Math%20inside%20RMarkdown,10n%3D1n2.>

Per: [Putting the value of a variable into a table in R Markdown, rather than it’s name - Stack Overflow](https://stackoverflow.com/questions/72902548/putting-the-value-of-a-variable-into-a-table-in-r-markdown-rather-than-its-nam)

Line 320 of this: <https://github.com/DARTH-git/cohort-modeling-tutorial-intro/blob/main/manuscript/cSTM_Tutorial_Intro.Rmd> is informative when viewing the final document on page 7 here: <https://arxiv.org/pdf/2001.07824.pdf>

This also includes within-cycle correction (WCC) using Simpson’s 1/3 rule, which I probably don’t want to bother applying, but at least I know it’s an option.

Some helpful tips on using math notation in R Markdown.

<https://rpruim.github.io/s341/S19/from-class/MathinRmd.html>

This paper says in technical terms which distributions fit and why:

Model Parameter Estimation and Uncertainty Analysis: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force Working Group–6 <file:///C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf>

Table X Model Parameters Values: Baseline, Ranges and Distributions for Sensitivity Analysis

| Parameter | Base Case Value | Minimum Value | Maximum Value | Source | Distribution |
| --- | --- | --- | --- | --- | --- |
| **Cost (Per Cycle)** |  |  |  |  |  |
| FOLFOX | c\_PFS\_Folfox | Minimum\_c\_PFS\_Folfox | Maximum\_c\_PFS\_Folfox |  | GAMMA(a.cIntervention\_c\_PFS\_Folfox, b.cIntervention\_c\_PFS\_Folfox) |
| FOLFIRI | c\_OS\_Folfiri | Minimum\_c\_OS\_Folfiri | Maximum\_c\_OS\_Folfiri |  | GAMMA(a.cIntervention\_c\_OS\_Folfiri, b.cIntervention\_c\_OS\_Folfiri) |
| Bevacizumab | c\_PFS\_Bevacizumab | Minimum\_c\_PFS\_Bevacizumab | Maximum\_c\_PFS\_Bevacizumab |  | GAMMA( a.cIntervention\_c\_PFS\_Bevacizumab, b.cIntervention\_c\_PFS\_Bevacizumab ) |
| Administration Cost | administration\_cost | Minimum\_administration\_cost | Maximum\_administration\_cost |  | GAMMA(a.cIntervention\_administration\_cost, b.cIntervention\_administration\_cost) |
| **Adverse Event Cost** |  |  |  |  |  |
| Leukopenia | c\_AE1 | Minimum\_c\_AE1 | Maximum\_c\_AE1 |  | GAMMA(a.cIntervention\_c\_AE1, b.cIntervention\_c\_AE1) |
| Diarrhea | c\_AE2 | Minimum\_c\_AE2 | Maximum\_c\_AE2 |  | GAMMA(a.cIntervention\_c\_AE2, b.cIntervention\_c\_AE2) |
| Vomiting | c\_AE3 | Minimum\_c\_AE3 | Maximum\_c\_AE3 |  | GAMMA(a.cIntervention\_c\_AE3, b.cIntervention\_c\_AE3) |
|  |  |  |  |  |  |
| **Adverse Event Incidence - With Bevacizumab** |  |  |  |  |  |
| Leukopenia | p\_FA1\_Exp | Minimum\_p\_FA1\_EXPR | Maximum\_p\_FA1\_EXPR |  | BETA(alpha\_p\_FA1\_EXPR, beta\_p\_FA1\_EXPR) |
| Diarrhea | p\_FA2\_Exp | Minimum\_p\_FA2\_EXPR | Maximum\_p\_FA2\_EXPR |  | BETA(alpha\_p\_FA2\_EXPR, beta\_p\_FA2\_EXPR) |
| Vomiting | p\_FA3\_Exp | Minimum\_p\_FA3\_EXPR | Maximum\_p\_FA3\_EXPR |  | BETA(alpha\_p\_FA3\_EXPR, beta\_p\_FA3\_EXPR) |
| **Adverse Event Incidence - Without Bevacizumab** |  |  |  |  |  |
| Leukopenia | p\_FA1\_STD | Minimum\_p\_FA1\_STD | Maximum\_p\_FA1\_STD |  | BETA(alpha\_p\_FA1\_STD , beta\_p\_FA1\_STD) |
| Diarrhea | p\_FA2\_STD | Minimum\_p\_FA2\_STD | Maximum\_p\_FA2\_STD |  | BETA(alpha\_p\_FA2\_STD , beta\_p\_FA2\_STD) |
| Vomiting | p\_FA3\_STD | Minimum\_p\_FA3\_STD | Maximum\_p\_FA3\_STD |  | BETA(alpha\_p\_FA3\_STD , beta\_p\_FA3\_STD) |
| **Utility (Per Cycle)** |  |  |  |  |  |
| Progression Free Survival | u\_F | Minimum\_u\_F | Maximum\_u\_F |  | BETA(u\_F\_alpha, u\_F\_beta) |
| Overall Survival | u\_P | Minimum\_u\_P | Maximum\_u\_P |  | BETA(u\_P\_alpha , u\_P\_beta) |
| **Adverse Event Disutility** |  |  |  |  |  |
| Leukopenia | AE1\_DisUtil | Minimum\_AE1\_DisUtil | Maximum\_AE1\_DisUtil | BRTYA | alpha\_u\_AE1, beta\_u\_AE1 |
| Diarrhea | AE2\_DisUtil | Minimum\_AE2\_DisUtil | Maximum\_AE2\_DisUtil | BETA | alpha\_u\_AE2, beta\_u\_AE2 |
| Vomiting | AE3\_DisUtil | Minimum\_AE3\_DisUtil | Maximum\_AE3\_DisUtil | BETA | alpha\_u\_AE3, beta\_u\_AE3 |
| **Hazard Ratios** |  |  |  |  |  |
| PFS to OS under the Experimental Strategy | HR\_FP\_Exp | Minimum\_HR\_FP\_Exp | Maximum\_HR\_FP\_Exp | (Smeets et al. 2018) | rlnorm() |
| OS to PFS under the Experimental Strategy | HR\_PD\_Exp | Minimum\_HR\_PD\_Exp | Maximum\_HR\_PD\_Exp | (Smeets et al. 2018) | rlnorm() |
| **Probability of Dying under Second-Line Treatment** | P\_OSD\_SoC | Minimum\_P\_OSD\_SoC | Maximum\_P\_OSD\_SoC |  | BETA(P\_OSD\_SoC\_alpha, P\_OSD\_SoC\_beta) |
| **Discount Rate** |  |  |  |  |  |
| Costs | d\_c | 0 | 0.08 |  |  |
| Outcomes | d\_e | 0 | 0.08 |  |  |

## 09.4.6 Scenario Analysis

The below study includes scenario analysis at the end:

Rivera, F., Valladares, M., Gea, S., & López-Martínez, N. (2017). Cost-effectiveness analysis in the Spanish setting of the PEAK trial of panitumumab plus mFOLFOX6 compared with bevacizumab plus mFOLFOX6 for first-line treatment of patients with wild-type RAS metastatic colorectal cancer. Journal of Medical Economics, 20(6), 574-584. <https://sci-hub.st/10.1080/13696998.2017.1285780> also saved here: C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Evidence Synthesis\Economic Models\Rivera et al\_2017\_Cost-effectiveness analysis in the Spanish setting of the PEAK trial of.pdf

Scenario Analysis: “Often, uncertainty in a parameter may be represented by several discrete values, instead of a continuous range, sometimes called scenario analyses (e.g., evidence from clinical studies, utility surveys, or cost data sets may lead to different values). It is acceptable to report alternative outcomes under each of these discrete assumptions to complement other uncertainty analysis.” - Per: Model Parameter Estimation and Uncertainty Analysis C:/Users/Jonathan/OneDrive%20-%20Royal%20College%20of%20Surgeons%20in%20Ireland/COLOSSUS/Briggs%20et%20al%202012%20model%20parameter%20estimation%20and%20uncertainty.pdf

## 09.4.7 Calibration

The following has a calibration section: Lawrence, D., Maschio, M., Leahy, K. J., Yunger, S., Easaw, J. C., & Weinstein, M. C. (2013). Economic analysis of bevacizumab, cetuximab, and panitumumab with fluoropyrimidine-based chemotherapy in the first-line treatment of KRAS wild-type metastatic colorectal cancer (mCRC). Journal of medical economics, 16(12), 1387-1398. <https://www.tandfonline.com/doi/pdf/10.3111/13696998.2013.852097?needAccess=true>

C:\Users\Jonathan\OneDrive - Royal College of Surgeons in Ireland\COLOSSUS\Training Resources\Model Calibration in R

## References:

Generated automatically when the document is knitted.

Useful Darth publications to cite when using this code:

* Jalal H, Pechlivanoglou P, Krijkamp E, Alarid-Escudero F, Enns E, Hunink MG. An Overview of R in Health Decision Sciences. Med Decis Making. 2017; 37(3): 735-746. <https://journals.sagepub.com/doi/abs/10.1177/0272989X16686559>
* Alarid-Escudero F, Krijkamp EM, Enns EA, Yang A, Hunink MGM Pechlivanoglou P, Jalal H. Cohort State-Transition Models in R: A Tutorial. arXiv:200107824v2. 2020:1-48. <http://arxiv.org/abs/2001.07824>
* Krijkamp EM, Alarid-Escudero F, Enns EA, Jalal HJ, Hunink MGM, Pechlivanoglou P. Microsimulation modeling for health decision sciences using R: A tutorial. Med Decis Making. 2018;38(3):400–22. <https://journals.sagepub.com/doi/abs/10.1177/0272989X18754513>
* Krijkamp EM, Alarid-Escudero F, Enns E, Pechlivanoglou P, Hunink MM, Jalal H. A Multidimensional Array Representation of State-Transition Model Dynamics. Med Decis Making. Online First <https://doi.org/10.1177/0272989X19893973>

Aballéa, Samuel, Jeremy V. M. Chancellor, Maria Raikou, Michael F. Drummond, Milton C. Weinstein, Sophia Jourdan, and John Bridgewater. 2007. “Cost-Effectiveness Analysis of Oxaliplatin Compared with 5-Fluorouracil/Leucovorin in Adjuvant Treatment of Stage III Colon Cancer in the US.” *Cancer* 109 (6): 1082–89. <https://doi.org/10.1002/cncr.22512>.

Alarid-Escudero, Fernando, Eline M. Krijkamp, Eva A. Enns, Alan Yang, M. G. Myriam Hunink, Petros Pechlivanoglou, and Hawre Jalal. 2021. “An Introductory Tutorial on Cohort State-Transition Models in r Using a Cost-Effectiveness Analysis Example.” *arXiv:2001.07824 [q-Bio, Stat]*, August. <http://arxiv.org/abs/2001.07824>.

Briggs, Andrew H., Milton C. Weinstein, Elisabeth AL Fenwick, Jonathan Karnon, Mark J. Sculpher, and A. David Paltiel. 2012. “Model Parameter Estimation and Uncertainty Analysis: A Report of the ISPOR-SMDM Modeling Good Research Practices Task Force Working Group6.” *Medical Decision Making* 32 (5): 722732.

Franken, M. D., E. M. van Rooijen, A. M. May, H. Koffijberg, H. van Tinteren, L. Mol, A. J. ten Tije, et al. 2017. “Cost-Effectiveness of Capecitabine and Bevacizumab Maintenance Treatment After First-Line Induction Treatment in Metastatic Colorectal Cancer.” *European Journal of Cancer* 75 (April): 204–12. <https://doi.org/10.1016/j.ejca.2017.01.019>.

Goldstein, Daniel A., Bilal B. Ahmad, Qiushi Chen, Turgay Ayer, David H. Howard, Joseph Lipscomb, Bassel F. El-Rayes, and Christopher R. Flowers. 2015. “Cost-Effectiveness Analysis of Regorafenib for Metastatic Colorectal Cancer.” *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology* 33 (32): 3727–32. <https://doi.org/10.1200/JCO.2015.61.9569>.

Goldstein, Daniel A., Qiushi Chen, Turgay Ayer, David H. Howard, Joseph Lipscomb, Bassel F. El-Rayes, and Christopher R. Flowers. 2015a. “First-and Second-Line Bevacizumab in Addition to Chemotherapy for Metastatic Colorectal Cancer: A United Statesbased Cost-Effectiveness Analysis.” *Journal of Clinical Oncology* 33 (10): 1112.

———. 2015b. “First- and Second-Line Bevacizumab in Addition to Chemotherapy for Metastatic Colorectal Cancer: A United Statesbased Cost-Effectiveness Analysis.” *Journal of Clinical Oncology* 33 (10): 1112–18. <https://doi.org/10.1200/JCO.2014.58.4904>.

Goldstein, Daniel A., Qiushi Chen, Turgay Ayer, David H. Howard, Joseph Lipscomb, R. Donald Harvey, Bassel F. El-Rayes, and Christopher R. Flowers. 2014a. “Cost Effectiveness Analysis of Pharmacokinetically-Guided 5-Fluorouracil in FOLFOX Chemotherapy for Metastatic Colorectal Cancer.” *Clinical Colorectal Cancer* 13 (4): 219–25. <https://doi.org/10.1016/j.clcc.2014.09.007>.

———. 2014b. “Cost Effectiveness Analysis of Pharmacokinetically-Guided 5-Fluorouracil in FOLFOX Chemotherapy for Metastatic Colorectal Cancer.” *Clinical Colorectal Cancer* 13 (4): 219–25. <https://doi.org/10.1016/j.clcc.2014.09.007>.

Goldstein, Daniel A., Qiushi Chen, Turgay Ayer, David H. Howard, Joseph Lipscomb, Suresh S. Ramalingam, Fadlo R. Khuri, and Christopher R. Flowers. 2015. “Necitumumab in Metastatic Squamous Cell Lung Cancer: Establishing a Value-Based Cost.” *JAMA Oncology* 1 (9): 12931300.

Goulart, Bernardo, and Scott Ramsey. 2011b. “A Trial-Based Assessment of the Cost-Utility of Bevacizumab and Chemotherapy Versus Chemotherapy Alone for Advanced Non-Small Cell Lung Cancer.” *Value in Health* 14 (6): 836–45. <https://doi.org/10.1016/j.jval.2011.04.004>.

———. 2011a. “A Trial-Based Assessment of the Cost-Utility of Bevacizumab and Chemotherapy Versus Chemotherapy Alone for Advanced Non-Small Cell Lung Cancer.” *Value in Health* 14 (6): 836–45. <https://doi.org/10.1016/j.jval.2011.04.004>.

Guan, Xin, Hongchao Li, Xiaomo Xiong, Cike Peng, Ning Wang, Xiao Ma, and Aixia Ma. 2021. “Cost-Effectiveness Analysis of Fruquintinib Versus Regorafenib as the Third-Line Therapy for Metastatic Colorectal Cancer in China.” *Journal of Medical Economics* 24 (1): 339–44. <https://doi.org/10.1080/13696998.2021.1888743>.

Hamdy Elsisi, Gihan, Yousery Nada, Noha Rashad, and João Carapinha. 2019. “Cost-Effectiveness of Sorafenib Versus Best Supportive Care in Advanced Hepatocellular Carcinoma in Egypt.” *Journal of Medical Economics* 22 (2): 163–68. <https://doi.org/10.1080/13696998.2018.1552432>.

Hochster, Howard S., Lowell L. Hart, Ramesh K. Ramanathan, Barrett H. Childs, John D. Hainsworth, Allen L. Cohn, Lucas Wong, et al. 2008. “Safety and Efficacy of Oxaliplatin and Fluoropyrimidine Regimens With or Without Bevacizumab As First-Line Treatment of Metastatic Colorectal Cancer: Results of the TREE Study.” *Journal of Clinical Oncology* 26 (21): 3523–29. <https://doi.org/10.1200/JCO.2007.15.4138>.

Jalal, Hawre, Petros Pechlivanoglou, Eline Krijkamp, Fernando Alarid-Escudero, Eva Enns, and M. G. Myriam Hunink. 2017. “An Overview of R in Health Decision Sciences.” *Medical Decision Making* 37 (7): 735–46. <https://doi.org/10.1177/0272989X16686559>.

Krijkamp, Eline M., Fernando Alarid-Escudero, Eva A. Enns, Hawre J. Jalal, M. G. Myriam Hunink, and Petros Pechlivanoglou. 2018. “Microsimulation Modeling for Health Decision Sciences Using R: A Tutorial.” *Medical Decision Making* 38 (3): 400–422. <https://doi.org/10.1177/0272989X18754513>.

Krijkamp, Eline M., Fernando Alarid-Escudero, Eva A. Enns, Petros Pechlivanoglou, M.G. Myriam Hunink, Alan Yang, and Hawre J. Jalal. 2020. “A Multidimensional Array Representation of State-Transition Model Dynamics.” *Medical Decision Making* 40 (2): 242–48. <https://doi.org/10.1177/0272989X19893973>.

Kristin, Erna, Dwi Endarti, Levina Chandra Khoe, Kartika Widayati Taroeno-Hariadi, Christina Trijayanti, Armansyah Armansyah, and Sudigdo Sastroasmoro. 2021. “Economic Evaluation of Adding Bevacizumab to Chemotherapy for Metastatic Colorectal Cancer (mCRC) Patients in Indonesia.” *Asian Pacific Journal of Cancer Prevention : APJCP* 22 (6): 1921–26. <https://doi.org/10.31557/APJCP.2021.22.6.1921>.

Mitchell, Mark. 2022. *Engauge Digitizer Version 12.2.* [Available from: http://digitizer.sourceforge.net](Available%20from:%20http://digitizer.sourceforge.net).

Nebuloni, Daniela R., Milena P. Mak, Fabiano Hahn Souza, Daniel F. Saragiotto, Thiago Júlio, Gilberto De Castro Jr, Jorge Sabbaga, and Paulo M. Hoff. 2013. “Modified FLOX as First-Line Chemotherapy for Metastatic Colorectal Cancer Patients in the Public Health System in Brazil: Effectiveness and Cost-Utility Analysis.” *Molecular and Clinical Oncology* 1 (1): 175–79. <https://doi.org/10.3892/mco.2012.12>.

Nwaokorie, Annabelle, and Dirk Fey. 2021. “Personalised Medicine for Colorectal Cancer Using Mechanism-Based Machine Learning Models.” *International Journal of Molecular Sciences* 22 (18): 9970. <https://doi.org/10.3390/ijms22189970>.

Ramsey, Scott D., M. Robyn Andersen, Ruth Etzioni, Carol Moinpour, Sue Peacock, Arnold Potosky, and Nicole Urban. 2000. “Quality of Life in Survivors of Colorectal Carcinoma.” *Cancer* 88 (6): 1294–1303. https://doi.org/[https://doi.org/10.1002/(SICI)1097-0142(20000315)88:6<1294::AID-CNCR4>3.0.CO;2-M](https://doi.org/10.1002/(SICI)1097-0142(20000315)88:6%3c1294::AID-CNCR4%3e3.0.CO;2-M).

Refaat, Tamer, Mehee Choi, Germaine Gaber, Krystyna Kiel, Minesh Mehta, William Gradishar, and William Small. 2014. “Markov model and cost-effectiveness analysis of bevacizumab in HER2-negative metastatic breast cancer.” *American Journal of Clinical Oncology* 37 (5): 480–85. <https://doi.org/10.1097/COC.0b013e31827e4e9a>.

Shih, Ted, and Celeste Lindley. 2006. “Bevacizumab: An Angiogenesis Inhibitor for the Treatment of Solid Malignancies.” *Clinical Therapeutics* 28 (11): 1779–1802. <https://doi.org/10.1016/j.clinthera.2006.11.015>.

Siegel, Rebecca L., Kimberly D. Miller, and Ahmedin Jemal. 2019. “Cancer Statistics, 2019.” *CA: A Cancer Journal for Clinicians* 69 (1): 734.

Smeets, Dominiek, Ian S. Miller, Darran P. O’Connor, Sudipto Das, Bruce Moran, Bram Boeckx, Timo Gaiser, Johannes Betge, Ana Barat, and Rut Klinger. 2018. “Copy Number Load Predicts Outcome of Metastatic Colorectal Cancer Patients Receiving Bevacizumab Combination Therapy.” *Nature Communications* 9 (1): 116.

Tiwari, Ankita, Shivani Saraf, Amit Verma, Pritish Kumar Panda, and Sanjay K. Jain. 2018. “Novel Targeting Approaches and Signaling Pathways of Colorectal Cancer: An Insight.” *World Journal of Gastroenterology* 24 (39): 4428.

Vabi, Benjamin W., John F. Gibbs, and Glenn S. Parker. 2021. “Implications of the Growing Incidence of Global Colorectal Cancer.” *Journal of Gastrointestinal Oncology* 12 (Suppl 2): S387–98. <https://doi.org/10.21037/jgo-2019-gi-06>.

Wang, Huijuan, Lingfei Huang, Peng Gao, Zhengyi Zhu, Weifeng Ye, Haiying Ding, and Luo Fang. 2020. “Cost-Effectiveness Analysis of Cetuximab Combined with Chemotherapy as a First-Line Treatment for Patients with RAS Wild-Type Metastatic Colorectal Cancer Based on the TAILOR Trial.” *BMJ Open* 10 (2): e030738. <https://doi.org/10.1136/bmjopen-2019-030738>.

Wen, Feng, Hanrui Zheng, Yifan Wu, John Wheeler, Xiaoxi Zeng, Ping Fu, and Qiu Li. 2016. “Cost-Effectiveness Analysis of Fluorouracil, Leucovorin, and Irinotecan Versus Epirubicin, Cisplatin, and Capecitabine in Patients with Advanced Gastric Adenocarcinoma.” *Scientific Reports* 6 (1): 36060. <https://doi.org/10.1038/srep36060>.